SLIIT UNI
THE KNOWLEDGE UNIVERSITY

IT4010 Research Project – 2025 July

Topic Assessment Form

Project ID : 25-26J-525

1. Topic (12 words max)

Emotion-Aware Agile System for Progress Tracking, Expertise Recommendation, Requirement Management, and Inclusive Communication

2. Research group the project belongs to

AIMS - Autonomous Intelligent Machines and Systems

3. Specialization of the project belongs to

Software Engineering (SE)

4. If a continuation of a previous project:

Project ID	
Year	

5. Brief description of the research problem including references (200 – 500 words max) – references not included in word count.

In Agile software development teams, communication via platforms like Slack, GitHub, or Jira often reflects not only technical collaboration but also the emotional state of team members. However, emotional signals such as frustration, burnout, or demotivation often go unnoticed, leading to reduced productivity, interpersonal conflict, and inefficient sprint planning. Traditional Agile systems lack emotional intelligence and are not responsive to human factors in team dynamics.

This research aims to address that gap by building an emotion-aware AI assistant that understands English communication from team members. The system will extract emotional cues using Natural Language Processing (NLP) and facial expression recognition. It will then analyze how these emotions correlate with Agile metrics (e.g., task completion rates, velocity, burnout risks) and propose smart improvements to the Agile process.

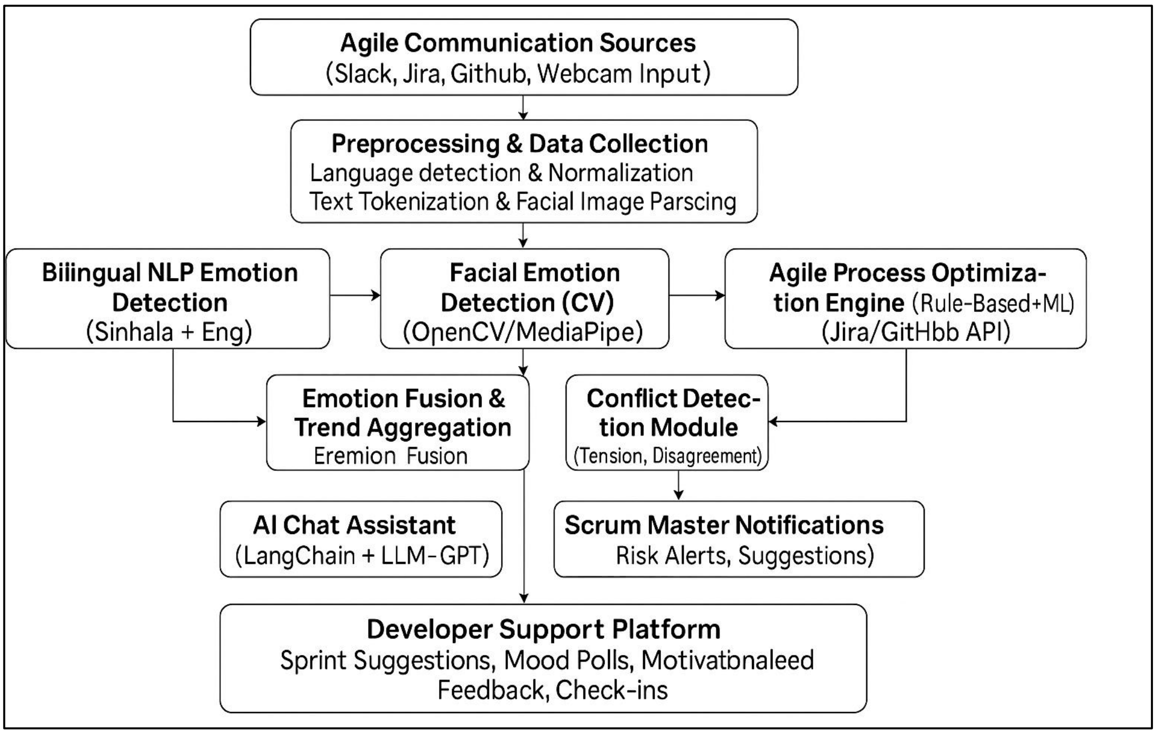
Beyond passive emotion tracking, the assistant will include an AI chatbot capable of giving decision-support suggestions (not just replies), using insights from emotion detection and process optimization. A final module will monitor communication patterns to proactively detect conflicts or tension and recommend interventions to Scrum Masters before productivity or morale is affected.

This project will combine multiple AI techniques (emotion classification, decision trees, LLMs, conflict modeling) to enhance Agile workflows in a way that is both intelligent and emotionally responsive — the first known system to combine these in English hybrid development teams.

6. Brief description of the nature of the solution including a conceptual diagram (250 words max)

The proposed solution is a multi-module AI assistant that continuously processes communication data from Agile development platforms. It consists of four main components:

1. Emotion Detection Engine: Uses NLP (BERT, RoBERTa) and facial recognition models to classify team members' emotions from Slack messages, GitHub comments, Jira tasks, and webcam inputs.
2. Agile Process Optimization Engine: Analyzes emotional trends and maps them to Agile metrics (e.g., task delays, sprint velocity). Uses rule-based logic and decision tree algorithms to recommend process changes like rescoping, inserting retrospectives, or reducing workload.
3. AI Chat Assistant + Decision Helper: A proactive bot integrated into chat tools that not only responds to team queries but also suggests task assignments, mood polls, or sprint planning changes based on real-time emotional and performance data.
4. Emotion-Aware Conflict Detection System: Monitors ongoing conversations and detects early signs of tension, negative exchanges, or emotional overload. It triggers warnings or recommends interventions like check-ins or private resolution prompts.


Agile Communication Sources
(Slack, Jira, Github, Webcam Input)
Preprocessing & Data Collection
Language detection & Normalization
Text Tokenization & Facial Image Parscing
Biiingual NLP Emotion Detection
(Sinhala + Eng)
Facial Emotion Detection (CV)
(OnenCV/MediaPipe)
Agile Process Optimiza-tion Engine (Rule-Based+ML)
(Jira/GitHbb API)
Emotion Fusion & Trend Aggregation
Eremion Fusion
Conflict Detec-tion Module
(Tension, Disagreement)
AI Chat Assistant
(LangChain + LLM-GPT)
Scrum Master Notifications
Risk Alerts, Suggestions)
Developer Support Platform
Sprint Suggestions, Mood Polls, Motivatbnaleed
Feedback, Check-ins

7. Brief description of specialized domain expertise, knowledge, and data requirements (300 words max)

This project requires interdisciplinary knowledge across **Natural Language Processing**, **Affective Computing**, **Agile Software Engineering**, and **Human-Computer Interaction**.

- **NLP expertise** is required to build bilingual emotion classifiers that analyze English texts. The team will work with pre-trained models (BERT, RoBERTa, XLM-R) and fine-tune them using custom datasets from developer chats and issue trackers.
- **Emotion recognition** also involves facial expression analysis using computer vision libraries like OpenCV and MediaPipe, combined with deep learning models to cross-verify text-based signals.
- **Agile domain knowledge** is critical to build meaningful links between emotional trends and Scrum metrics such as velocity, work-in-progress (WIP), sprint completion, and team burnout.
- The **LLM-based AI Assistant** will involve prompt engineering and integration with GPT-4 or open-source LLMs using LangChain. It must understand Agile-specific queries and respond with helpful suggestions based on ongoing team conditions.
- For the **conflict detection module**, research into team behavior, workplace psychology, and interpersonal communication patterns will guide the model design.

Data Requirements:

- Slack or Jira text logs
- Annotated emotion datasets (EmotionLines, EmoReact)
- GitHub commit messages
- Optional: synthetic datasets for facial emotion + conversational conflict

8. Objectives and Novelty

Main Objective To build an emotion-aware AI assistant that helps Agile development teams work more efficiently and collaboratively by analyzing emotional signals from communication and suggesting process improvements or conflict resolutions.			
Member Name with Registration No	Sub Objective	Tasks	Novelty
Fonseka A.A.T.N. IT22126092	Create agile Emotion Monitoring System	**Messaging Analysis** • Reads messages in our platform during daily standups, planning, or retrospectives. • Detects emotions like anger, love, fear, joy, sadness, surprise from what people type. • Tracks how emotions change over time in the conversation. **Facial & Micro-Expression Recognition** • Uses the webcam during video meetings to detect real-time facial expressions. • Captures micro-expressions (very short expressions under 0.5 seconds) that reveal hidden stress or tension.	1. **Multimodal Emotion Detection** Most existing tools either focus on text analysis (like sentiment in Slack messages) or facial expressions (like Affectiva or Microsoft Emotion API). Our system combines multiple modalities: text, facial micro-expressions, and voice tone. This fusion approach improves accuracy, especially when one modality is unclear (e.g., low-quality video or ambiguous text). Novelty: Integrating multiple streams of information in real-time to reliably detect nuanced emotions in Agile team settings. 2. **Micro-Expression Recognition** Detecting micro-expressions (<0.5 seconds) is challenging and rarely implemented in workplace tools. Capturing

		• Aggregates emotions across team members to understand the overall team mood. **Multimodal Fusion** • Combines text-based emotions and facial expressions to get a more reliable understanding. • If video is unclear, it relies more on text; if text is vague, it relies more on facial expressions. • Use voice tone for richer emotion detection. **Temporal Modeling** • Tracks how emotions evolve over time, across multiple meetings or sprints. • Predicts future emotional states, like whether a team member may feel frustrated or disengaged in upcoming meetings. **Sprint Velocity & Backlog Integration** • Links emotional trends with sprint velocity (amount of work completed), backlog distribution	these tiny, unconscious expressions allows detection of hidden stress, tension, or disengagement that people may not express verbally. Novelty: Real-time micro-expression detection in team meetings is not common in standard project management or communication tools. 3. **Temporal Emotion Modeling** Beyond snapshot detection, our system tracks emotional evolution over multiple meetings or sprints. Predicting future emotional states (like potential frustration or disengagement) is a proactive approach, unlike static sentiment dashboards. Novelty: The ability to anticipate emotional trends and link them to team dynamics is a forward-looking, predictive feature. 4. **Integration with Agile Metrics** Our system connects emotions to sprint velocity and backlog distribution. For example, it can detect patterns like: high frustration + low velocity → workload issues. Existing tools rarely tie emotional data

		(how tasks are spread among team members), and blocker resolution time (how quickly obstacles are removed). • Tracks participation and engagement levels in ceremonies, showing how hesitation or disengagement may affect team collaboration. • Highlights patterns such as rising frustration with declining velocity, uneven backlog assignments linked to stress, or hesitation correlating with delayed blocker resolution. • Provides decision support for Agile teams by showing how collective emotional states influence productivity, workload balance, and team engagement. **Emotion Confidence Scoring** • Each detection comes with a confidence score to ensure reliability. • Aggregates scores to team-level dashboards for Scrum Masters to make informed decisions	directly to performance metrics in Agile workflows. Novelty: Provides actionable insights for Agile decision-making, not just emotion visualization. 5. **Emotion Confidence Scoring** Each prediction comes with a confidence score, which is aggregated at the team level. This prevents overreaction to a single person’s emotional spike and ensures reliable interpretation. Novelty: Combining predictive confidence with aggregated team-level analysis is unique in workplace emotion systems. **6. Context-Aware Multimodal Fusion** The system dynamically adjusts reliance on different modalities: Video unclear → rely more on text Text ambiguous → rely more on facial expressions Many existing multimodal systems do not account for quality-dependent weighting in real-time. Novelty: Context-sensitive fusion makes emotion detection robust in real-world meeting conditions.

		without misinterpreting individual emotions.	
Amarasinghe C.N. IT22115348	AI-Based Expertise Recommendation System	**Data Collection & Integration** • Connect to Jira API to pull past issues, tickets, and their assignees. • Connect to GitHub API to pull commits, PRs, authors, and changed files. • Past Contributions (Bug/Ticket history) **Data Preprocessing & Feature Engineering** • Clean text data: remove stopwords, code snippets, and irrelevant metadata. • Convert task titles/descriptions into embeddings (BERT/CodeBERT). • Generate developer profile embeddings: average embeddings of tasks solved previously. • Compute match features: similarity between task and developer embeddings, tag	Integrating advanced semantic embeddings of Jira issues, GitHub commits, and PR data with developer historical profiles, combined with task-to-developer similarity scoring, and enhanced with ranking models to suggest the most suitable experts in Agile teams. **What makes it novel:** • Unlike existing task assignment or search tools that rely on keyword matching or manual lookup, Our system uses semantic embeddings to understand the meaning of tasks and developer expertise, not just text. • It doesn't just suggest a single assignee — it produces Top-3 ranked expert recommendations with confidence scores, improving

		overlap, file path overlap, language match. **Model Training** • network to produce a ranking of developers. Train with historical labels: who resolved each task/PR. **Continuous Learning / Feedback** • Log real task assignments vs model recommendations. • Retrain periodically to include new developers, projects, or changing expertise. • Optional human-in-the-loop feedback: let Scrum Master approve and refine recommendations.	assignment accuracy and workload balancing. • Introduces a developer-specific expertise representation (e.g., “authentication specialist”, “frontend UI expert”, “payments module contributor”) rather than generic labels, making it highly relevant and actionable for Agile project allocation. • Independent and research-oriented: can operate without relying on emotion detection or task redistribution components, making it a standalone intelligence module for knowledge sharing and expertise recommendation. • Supports cold-start scenarios for new developers or modules by initializing embeddings from team averages and semantic similarity, ensuring reliable

			recommendations even with sparse historical data.
Vithana P.I IT22136688	Requirement Change Impact & Sprint Replanning Tracker	**Backlog & Requirement Data Collection** • Collects sprint backlog and requirement data from Jira, Trello, or DevOps tools. • Tracks initial sprint plans, requirement changes, and sprint outcomes such as velocity, task completion, defects, and delivery delays.	Novelty is not only tracks changes in sprint backlog and requirements but also predicts their impact on key Agile metrics and provides proactive, data-driven recommendations for dynamic sprint replanning. • **Integrated Impact Prediction:** Unlike standard Agile tools (Jira, Trello) that only record changes and completed tasks, this component **estimates how**

		• Preprocesses data into a structured format for analysis. **Change Detection Module** • Automatically identifies added, removed, or modified backlog items when new sprint data is input. • Flags tasks that have changed story points, priority, or dependencies. • Maintains a history of changes to support impact analysis. **Feature Engineering** • Derive quantitative features linked to the 4 factors: 1. **Effort & Time** → Story points added/removed, velocity shifts, workload per developer. 2. **Schedule Risk** → Dependency changes, timing of changes, spillovers. 3. **Quality Impact** → Reopened issues, defect injection rate, failed tests.	**changes will affect velocity, completion rates, defects, delivery delays, and team workload** before the sprint ends. • **Proactive Decision Support:** It **recommends mitigation strategies** (splitting stories, reassigning tasks, adding buffers) rather than just reporting issues, enabling managers to act before problems occur. • **Dynamic Sprint Replanning:** Most existing tools do not automatically suggest **task redistribution or handle spillovers** when requirements change mid-sprint. This component enables **adaptive sprint management** based on predictive analytics. • **Data-Driven Insights:** Combines historical sprint data, change detection, and machine learning to provide **quantitative, actionable insights**, which goes beyond

		4. **Team Productivity/Morale** → Backlog churn, work reassignments, unfinished tasks. • Encode categorical features (e.g., priority levels, dependency presence) for model training. • Prepares structured data for impact prediction. **Impact Prediction Model** • Predict the effect of requirement changes on: 1. Sprint velocity (Effort/Time factor). 2. Probability of delivery delays (Schedule Risk factor). 3. Defect likelihood & product quality degradation (Quality factor). 4. Team productivity/morale risk (Productivity factor). • Trains and evaluates the model using historical sprint data to ensure accuracy, precision, recall, and F1-score (like predicting sprint success/failure, or detecting duplicates).	traditional dashboards or static reporting.

		Priority Scoring Engine • Combine the predicted impacts from the 4 factors into a single **priority score** for each backlog item. • Allow configurable weights so teams can emphasize what matters most (e.g., quality vs. schedule). • Continuously update priority scores whenever new requirement changes are detected. • Rank backlog items dynamically so that high-impact changes are highlighted for sprint replanning. **Proactive Risk Mitigation** • Recommend strategies to minimize negative impacts of requirement changes: 1. **Effort/Time Risk** → Split large stories into smaller ones.	

		2. **Schedule Risk** → Add buffers or re-sequence dependent tasks. 3. **Quality Risk** → Insert extra test cycles or technical spikes. 4. **Productivity Risk** → Reallocate resources or balance workload. • Highlight high-risk backlog items for Product Owner and Scrum Master attention. **Dynamic Sprint Replanning** • Generate revised sprint backlogs when high-impact changes are detected. • Suggest strategies: ○ Replace or postpone low-priority items. ○ Redistribute workload across team members. ○ Prioritize high-dependency items early. ○ Add extra testing capacity if quality risk is high.	

		• Allow the Scrum Master and Product Owner to **manually adjust and finalize the backlog** based on the system’s recommendations, ensuring team consensus and alignment with sprint goals. **Decision-Support Dashboard** • Visualize requirement changes and their predicted factor impacts. • Display risk metrics such as velocity drop %, defect increase %, probability of delivery delays, and productivity drop likelihood. • Show calculated priority scores and recommended sprint adjustments. • Provide drill-down insights into each factor (Effort, Schedule, Quality, Productivity **User-Friendly Outputs**	

		- Generate clear explanations for why a replanning recommendation was made (e.g., “High schedule risk due to dependency shift”). - Provide exportable reports (PDF/CSV) for sprint review and stakeholder communication. - Support real-time dashboard updates during sprint execution.	
M.B.H. De Silva IT22602732	Inclusive Brainstorming & Communication Empowerment Platform	Multi-Mode Idea Capture - Provide silent submission box (text, speech-to-text). - Support optional anonymity for contributors. - Detect hesitation signals (pauses, backspaces, aborted inputs). Rephraser & Clarity Enhancer - Grammar correction and rephrasing of fragmented inputs. - Preserve technical entities (IDs, numbers, terms). - Offer tone adjustment (polite/neutral/assertive). Confidence-Building Preview - Provide “before vs. after” preview of rephrased input.	- Moves beyond traditional brainstorming tools (Miro, Jamboard) by ensuring **shy or hesitant members** can contribute without pressure. - Unlike Grammarly/DeepL, this rewriter is **entity-preserving** and tailored for **Agile-style idea expression**. - First brainstorming system to act as a **confidence trainer**—users gain trust in their ideas before sharing. - Goes beyond idea capture by **structuring contributions** for fair group decision-making. - Most brainstorming tools **do not organize raw inputs**—this adds

		• Allow user edits and approval before posting. • Offer tone-control slider to boost assertiveness if desired. Brainstorming Idea Pool • Collect all polished ideas in a shared board (sticky-note style). • Enable team voting, commenting, and grouping of ideas. • Support anonymity in voting to reduce bias. Clustering & Summarization • Detect duplicates and semantically similar ideas. • Cluster related suggestions together. • Generate concise summaries of top ideas. Participation Diversity Analytics • Track number of unique contributors. • Show contribution balance (active vs. silent users). • Visualize participation equality.	intelligent clustering for efficient decision support.

9. Individual component description of how it is complied with the specialization.

Member Name with Registration No	Description
Fonseka A.A.T.N. IT22126092	This component captures and analyzes emotional signals from two main sources — text chat from the custom Agile meeting platform and commit messages from GitHub, alongside real-time facial expressions detected during video meetings. It uses NLP models fine-tuned on English developer communication to classify emotions such as “frustrated,” “blocked,” or “motivated.” Facial expression analysis includes high-frequency micro-expression detection to capture subtle stress indicators. A Hierarchical Attention Network dynamically adjusts the weight given to text vs. facial cues based on signal quality. Long Short-Term Memory (LSTM) models track and predict emotional changes over time, and each detection includes a confidence score to guide downstream decision-making.
Amarasinghe C.N. IT22115348	This component focuses on intelligently recommending the most suitable developer(s) for new tasks or issues raised in Agile teams. The system analyzes task descriptions from Jira, GitHub commits, or project documentation and matches them with developer expertise profiles built from their historical contributions, commit history, pull requests, and code ownership patterns. Using semantic embeddings and ranking algorithms, the system generates Top-N developer recommendations with confidence scores, helping project managers and Scrum Masters assign work more effectively. It is designed to be independent and research-oriented, operating without dependency on other emotion detection or redistribution components. It primarily aims to streamline knowledge sharing, reduce task assignment delays, and ensure tasks are allocated to the most qualified team members.
Vithana P.I IT22136688	This component collects and analyzes sprint backlog and requirement data from tools like Jira, Trello, or DevOps platforms. It tracks any added, removed, or changed tasks, calculates key metrics to understand sprint progress, and predicts how changes may affect velocity, task completion, defects, delivery delays, and team workload. It provides a dashboard showing risks and recommended actions, and helps dynamically replan the sprint by suggesting task reassignment, handling spillovers, and prioritizing dependencies for smoother, data-driven sprint management.
M.B.H. De Silva IT22602732	This component enhances inclusivity in Agile meetings by transforming hesitant, fragmented, or multilingual inputs into clear and professional communication. It supports multiple input modes: silent text submissions, speech-to-text for hesitant speakers, and multilingual inputs in Sinhala, Tamil,

	or English code-mixing. Natural Language Processing (NLP) models fine-tuned for grammar correction, rephrasing, and multilingual translation ensure that ideas are preserved while being expressed fluently. An entity-preserving mechanism safeguards critical technical terms, identifiers, and numerical values during rephrasing. A tone-control layer refines the communication style, allowing outputs to be assertive yet polite, aligned with Agile team norms. To address communication anxiety, a hesitation detection module monitors typing patterns, long silences, and aborted messages, prompting users with gentle encouragement or an offer to rephrase their input. A preview interface allows members to privately review and approve AI-generated outputs before sharing, giving them agency and confidence. By integrating these features, the component not only improves clarity and professionalism of team contributions but also ensures equal participation, unlocking valuable ideas from team members who might otherwise remain unheard.

10. Supervisor details

	Title	First Name	Last Name	Signature
Supervisor	Ms. (Lecturer)	Ishara	Weerathunga	
Co-Supervisor	Ms. (Lecturer)	Hansi	De Silva	
External Supervisor				
Summary of external supervisor’s (if any) experience and expertise				

This part is to be filled by the Topic Screening Staff members.

a) Does the chosen research topic possess a comprehensive scope suitable for a final-year project?

Yes		No	

b) Does the proposed topic exhibit novelty?

Yes		No	

c) Do you believe they have the capability to successfully execute the proposed project?

Yes		No	

d) Do the proposed sub-objectives reflect the students' areas of specialization?

Yes		No	

e) Supervisor's Evaluation and Recommendation for the Research topic:

Acceptable: Mark/Select as necessary

IT4010 Research Project – 2025 July

Topic Assessment Form

Topic Assessment Accepted	
Topic Assessment Accepted with minor changes*	
Topic Assessment to be Resubmitted with major changes*	
Topic Assessment Rejected. Topic must be changed	

* Detailed comments given below

Comments

Staff Member's Name	Signature

***Important**:

1. According to the comments given by the evaluator, make the necessary modifications and get the approval by the **Evaluator**.
2. If the project topic is rejected, identify a new topic, and request the RP Team for a new topic assessment.