**COSMOS** — Coordinated Orbital System Management and Operations Suite

**COSMOS Sprint Report**  
Sprint 1 Ending April 01, 2025  
Sprint 2 Ending April 22, 2025 Sprint 3 Ending May 05, 2025

**Context**

**First Day of Sprint 1:** March 25, 2025  
**Last Day of Sprint 1:** April 01, 2025  
**First Day of Sprint 2:** April 8, 2025  
**Last Day of Sprint 2:** April 22, 2025 **First Day of Sprint 3:** April 28, 2025 **Last Day of Sprint 3:** May 07, 2025   
**Working Days in Sprint 1:** 07  
**Working Days in Sprint 2:** 14 **Working Days in Sprint 3:** 09

**Team Members**

| **Name** | **Role in the Sprint** | **Planned Days** | **Worked Days** |
| --- | --- | --- | --- |
| Pasupuleti Jaswanth | Scrum Master/ Security Manager | 30 | 26 |
| Krithin Thota | Product Owner , Head developer | 30 | 28 |
| Dhanush Dayanand | Head Vision developer | 30 | 28 |
| Vangapandu Baladitya | Data Analyst, Vision Developer | 30 | 29 |

**Sprint 1, Sprint 2, Sprint 3**

**Product Owner:** Krithin Thotha **Scrum Master:** Pasupuleti Jaswanth   
**Internal Assessor:** Dr. S Santhanalakshmi

**1. User Stories**

**Sprint 1**

| **User Story ID** | **As a** | **I want to** | **So that I can** |
| --- | --- | --- | --- |
| 1 | System Architect | define the architecture for CS LLM, CV LLM, and Central LLM | Better understand working of the systems. |
| 2 | Project Team Member | communication protocols to be defined | Interact with different domains seamlessly |
| 3 | Developer | set up project repositories | Easily access code and update versions |

**Sprint 2**

| **User Story ID** | **As a** | **I want to** | **So that I can** |
| --- | --- | --- | --- |
| 4 | LLM Developer | Convert task updates into structured prompts | Improve communication between LLMs |
| 5 | Security Specialist | Implement quantum encryption | Protected inter-LLM communication is |
| 6 | CV Developer | convert detections into structured prompts | Pass information to LLM for further processing. |
| 7 | Data Architect | Establish data partitioning | Retrieve data and optimize |

**Sprint 3**

| **User Story ID** | **As a** | **I want to** | **So that I can** |
| --- | --- | --- | --- |
| 8 | System Operator | see the predicted path or impact area for detected space debris | visually understand the potential risks. |
| 9 | System Administrator | Deploy a UI | Oversee system activity |
| 10 | Developer | Implement feedback loops | continuously improve system |
| 11 | QA Engineer | Evaluate the CS LLMs | Improve communication delays |
|  |  |  |  |

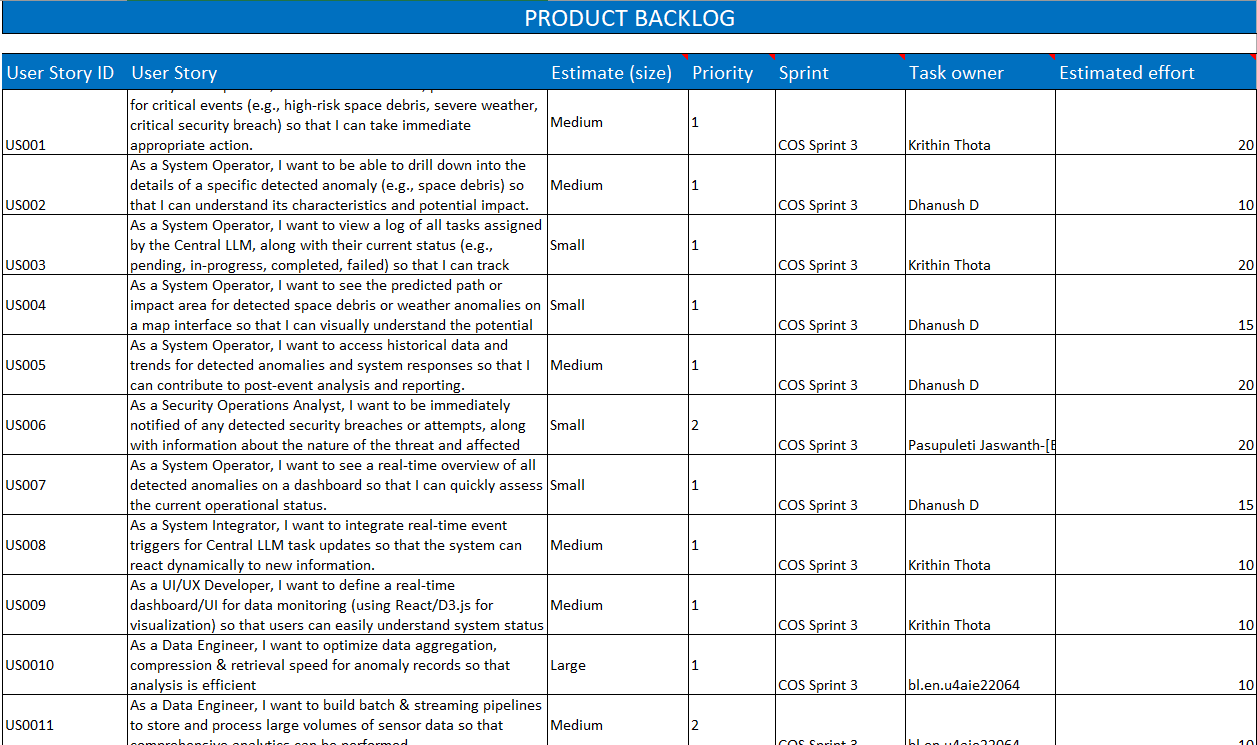
**2. Product Backlogs**

**Sprint 1 Product Backlog:**

|  |
| --- |
| * Project setup |
| * Define architecture for CS LLM , CV LLM , Central LLM |
| * setup repositories |
| * define communication protocols |
| * Data collection and preprocessing |
| * Gather dataset for CV LLM (space debris, weather anomalies) |
| * Define data encryption methods for CS LLM ( Quantum Security Framework) |
| * Gather dataset for task management to train LLMs ( Sub and central nodes) |
| * Model selection and baseline testing |
| * Select CV model |
| * Finalise and apply encryption methods for secure task transmission |
| * Research and Documentation |
| * Finalise prompt format for central LLM |
| * Document project scope and workflows |

**Sprint 2 Product Backlog:**

|  |
| --- |
| * Implement CV LLM |
| * Train object detection and prediction model for debris/weather anomalies. |
| * Implement thresholding for severity scoring. |
| * Convert detections into structured prompts for Central LLM. |
| * Implement CS LLM |
| * Implement quantum encryption for secure message relay. |
| * Establish message integrity verification. |
| * Convert task updates into structured prompts for Central LLM. |
| * Central LLM Development |
| * Implement multi-source prompt processing. |
| * Develop task prioritization algorithm based on severity & urgency. |
| * Implement reinforcement learning for task optimization. |
| * Integration of Sub-LLMs |
| * Establish real-time data flow from CS LLM & CV LLM to Central LLM. |
| * Set up APIs for task assignment. |
| **Sprint 3 Product Backlog:**   |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | |  | | --- | | * Performance Testing | | * Stress-test CV LLM on high-speed debris detection | | * Evaluate CS LLMs latency under encrypted messaging. | | * Optimize Central LLMs decision-making loop. | | * Final Integration & Refinement | | * Implement feedback loops to refine model accuracy. | | * Optimize task allocation efficiency.  |  | | --- | | * Real-Time task analytics | | * Implement Kafka/Spark Streaming to process CV anomaly detection data in real time. | | * Build batch & streaming pipelines to store and process large volumes of sensor data. | | * Optimize data aggregation, compression & retrieval speed for anomaly records. | | * Define real-time dashboard/UI for data monitoring (React/D3.js for visualization) | | * Integrate real-time event triggers for Central LLM task updates. | | * Implement data pipelines to feed Central LLM with processed insights from CV & CS LLMs. | | * Develop predictive analytics module using historical anomaly trends. | | * Test Big Data streaming performance under extreme loads. | | * Deploy UI for real-time monitoring of detected anomalies & task assignments. | | * Set up logging & monitoring tools for post-deployment analysis. | | | |  | |

****

**3. Iteration Planning**

**Vision and Roadmap:** Build an integrated task management suite for satellite automation

**Number of User Stories Fit:** 3 User Stories per sprint.

**Iteration Theme / Name:**

* Sprint 1: "Getting Started"
* Sprint 2: "Implementation and integration"
* Sprint 3: “Performance Testing”

**Dates:**

* Sprint 1: March 25–April 01, 2025
* Sprint 2: April 08–April 22, 2025
* Sprint 3: April 28–May 07-, 2025

**Capacity:**

* Sprint 1: 100 hours
* Sprint 2: 95 hours

**Definition of Done (DoD):**

* Feature fully developed, reviewed, tested, and deployed on staging.

**Dependencies and Assumptions:**

* UI library setup before frontend development.

**Commitment:**

* Complete 3 stories per sprint.

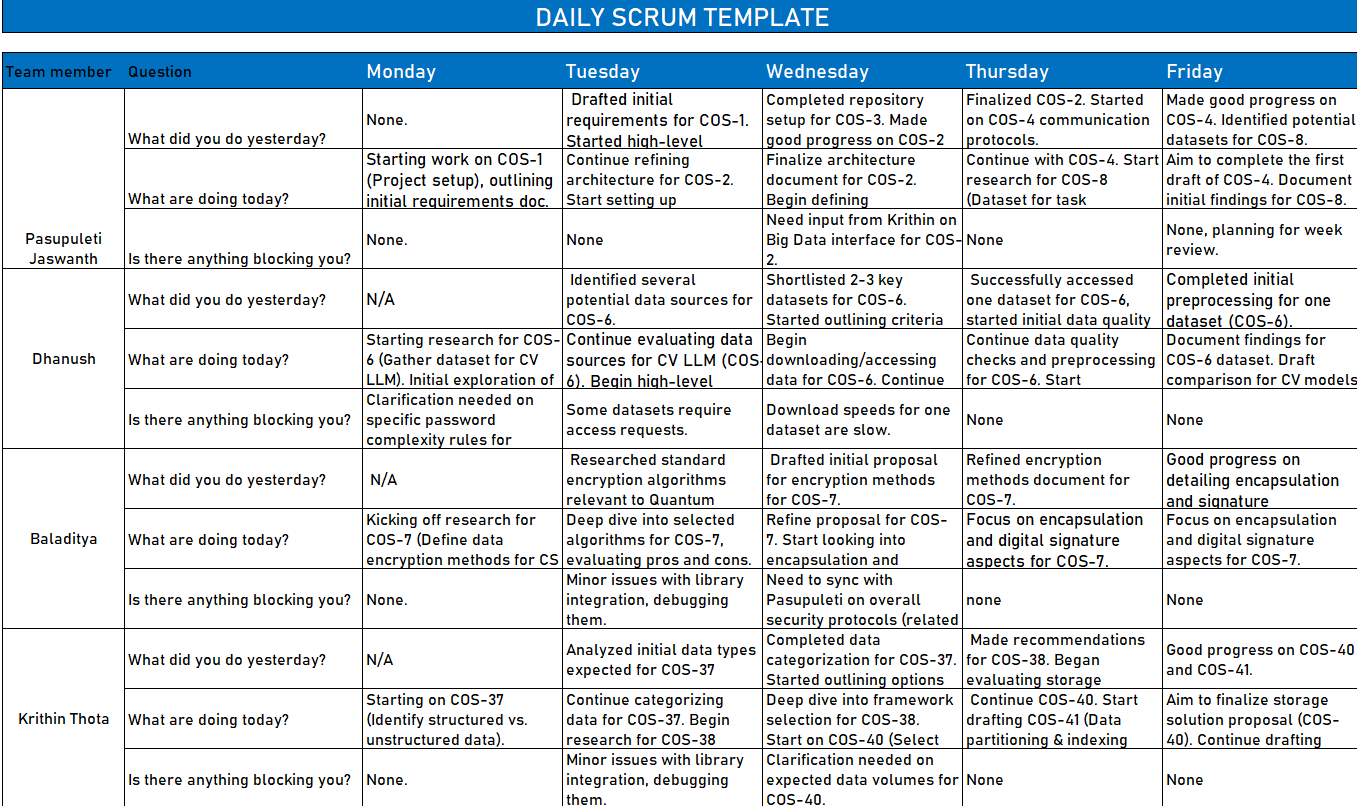
**Actions:**

* Daily scrums, code reviews, staging deployments.

**4.Daily Scrum**

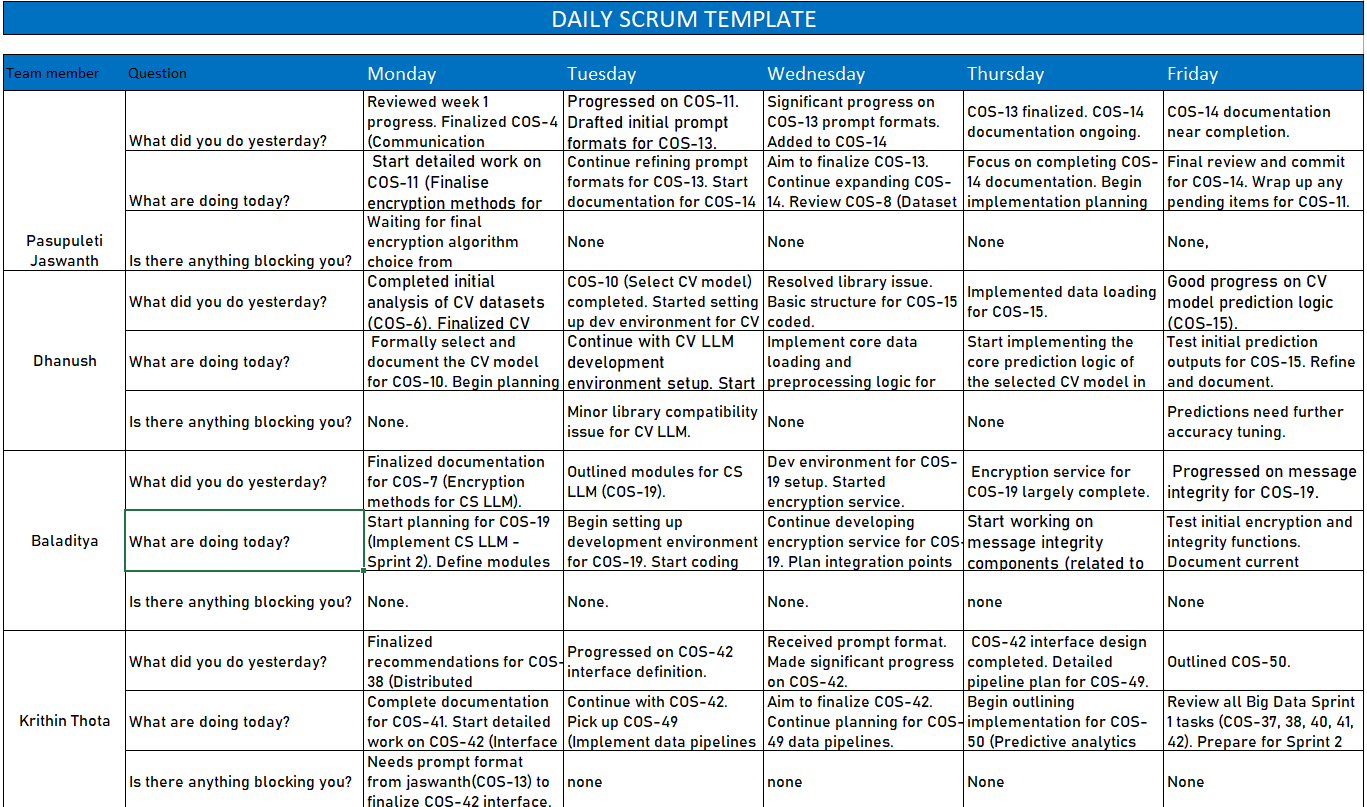
**Week 1 Review**

* Focus: Foundational project setup and initial research across all domains.
* Key Activities & Progress:
  + Pasupuleti Jaswanth:
    - Initiated project requirements (COS-1) and defined high-level architecture (COS-2).
    - Set up repositories (COS-3) and started on communication protocols (COS-4).
  + Dhanush:
    - Began gathering and performing initial quality checks on CV LLM datasets (COS-6).
    - Started outlining criteria for CV model selection (COS-10).
  + Baladitya:
    - Commenced research and drafting proposals for CS LLM data encryption methods (COS-7), including quantum security aspects and digital signatures.
  + Krithin Thota:
    - Worked on identifying data types (COS-37).
    - Evaluated storage solutions (COS-40) and drafted data partitioning strategies (COS-41).

****

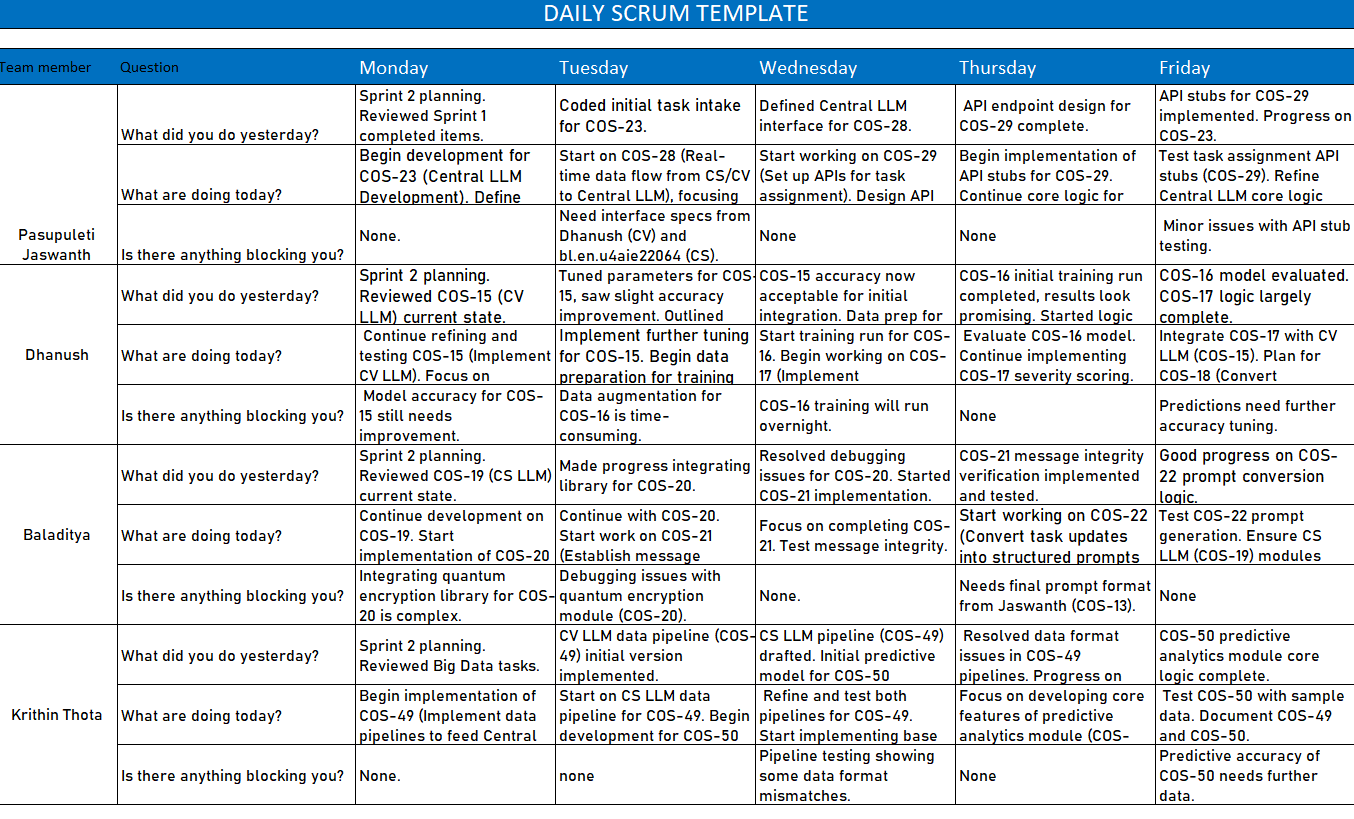
**Week 2 Review**

* **Focus: Transition from initial research to finalizing foundational documents and beginning development environment setups.**
* **Key Activities & Progress:** 
  + **Pasupuleti Jaswanth:** 
    - **Finalized communication protocols (COS-4).**
    - **Worked on finalizing encryption methods for task transmission (COS-11) and prompt formats (COS-13).**
  + **Dhanush:** 
    - **Completed CV model selection (COS-10).**
    - **Began setting up the CV LLM (COS-15) development environment, coding basic structures, and data loading logic.**
  + **Baladitya:** 
    - **Finalized CS LLM encryption methods documentation (COS-7).**
    - **Started planning and setting up the environment for the CS LLM (COS-19), including core encryption/decryption services and message integrity components.**
  + **Krithin Thota:** 
    - **Finalized recommendations for the distributed framework (COS-38) and storage (COS-40).**
    - **Made significant progress on defining the Big Data interface to the Central LLM (COS-42).**

****

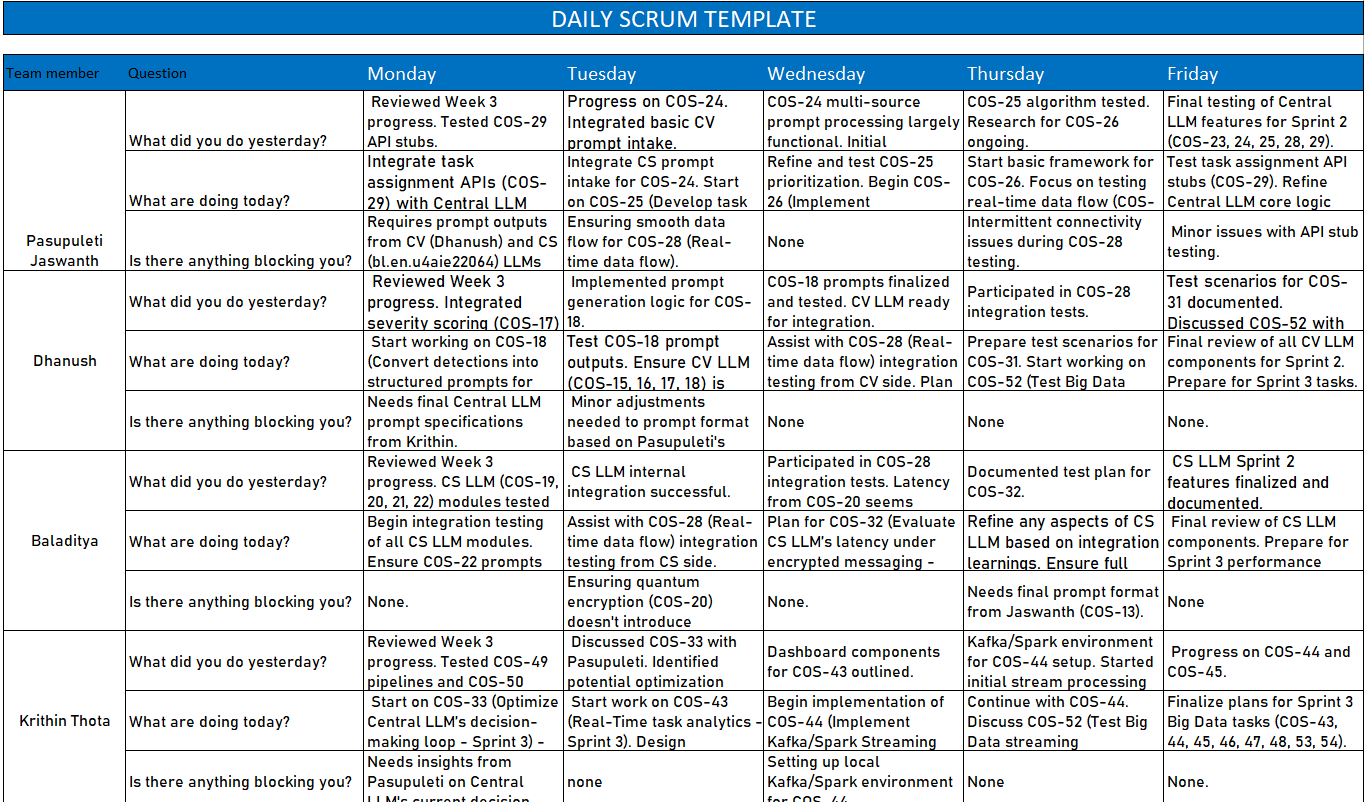
**Week 3 Review**

* **Focus: Core development for Sprint 2 tasks, emphasizing building LLM components and data pipelines.**
* **Key Activities & Progress:** 
  + **Pasupuleti Jaswanth:** 
    - **Began development on Central LLM task intake (COS-23).**
    - **Defined the Central LLM interface for real-time data flow (COS-28).**
    - **Started API setup for task assignment (COS-29), including design and stub implementation.**
  + **Dhanush:** 
    - **Focused on improving CV LLM (COS-15) accuracy.**
    - **Prepared data and trained the object detection model (COS-16), and evaluated it.**
    - **Started implementation of severity scoring (COS-17).**
  + **Baladitya:** 
    - **Continued CS LLM (COS-19) development.**
    - **Worked on quantum encryption (COS-20), message integrity (COS-21), and prompt conversion (COS-22).**
  + **Krithin Thota:** 
    - **Implemented data pipelines for CV and CS LLM data to feed the Central LLM (COS-49).**
    - **Developed base logic and core features for the predictive analytics module (COS-50).**

****

Week 4 Review

* Focus: Integration efforts, testing of Sprint 2 features, and planning for Sprint 3.
* Key Activities & Progress:
  + Pasupuleti Jaswanth:
    - Tested real-time data flow (COS-28).
  + Dhanush:
    - Finalized prompt generation for the CV LLM (COS-18).
    - Planned for stress-testing the CV LLM (COS-31) and Big Data streaming performance (COS-52).
  + Baladitya:
    - Completed CS LLM module integration (COS-19, 20, 21, 22).
    - Participated in COS-28 integration testing, noting acceptable latency from quantum encryption (COS-20).
  + Krithin Thota:
    - Tested pipelines (COS-49) and the predictive module (COS-50).
    - Began research for optimizing the Central LLM’s decision loop (COS-33).
    - Started implementing Kafka/Spark streaming (COS-44) and planned other Sprint 3 Big Data tasks (e.g., COS-45)

****

**5. Test cases**

**Test Cases for Sprint 1**

**Project Name: COSMOS**

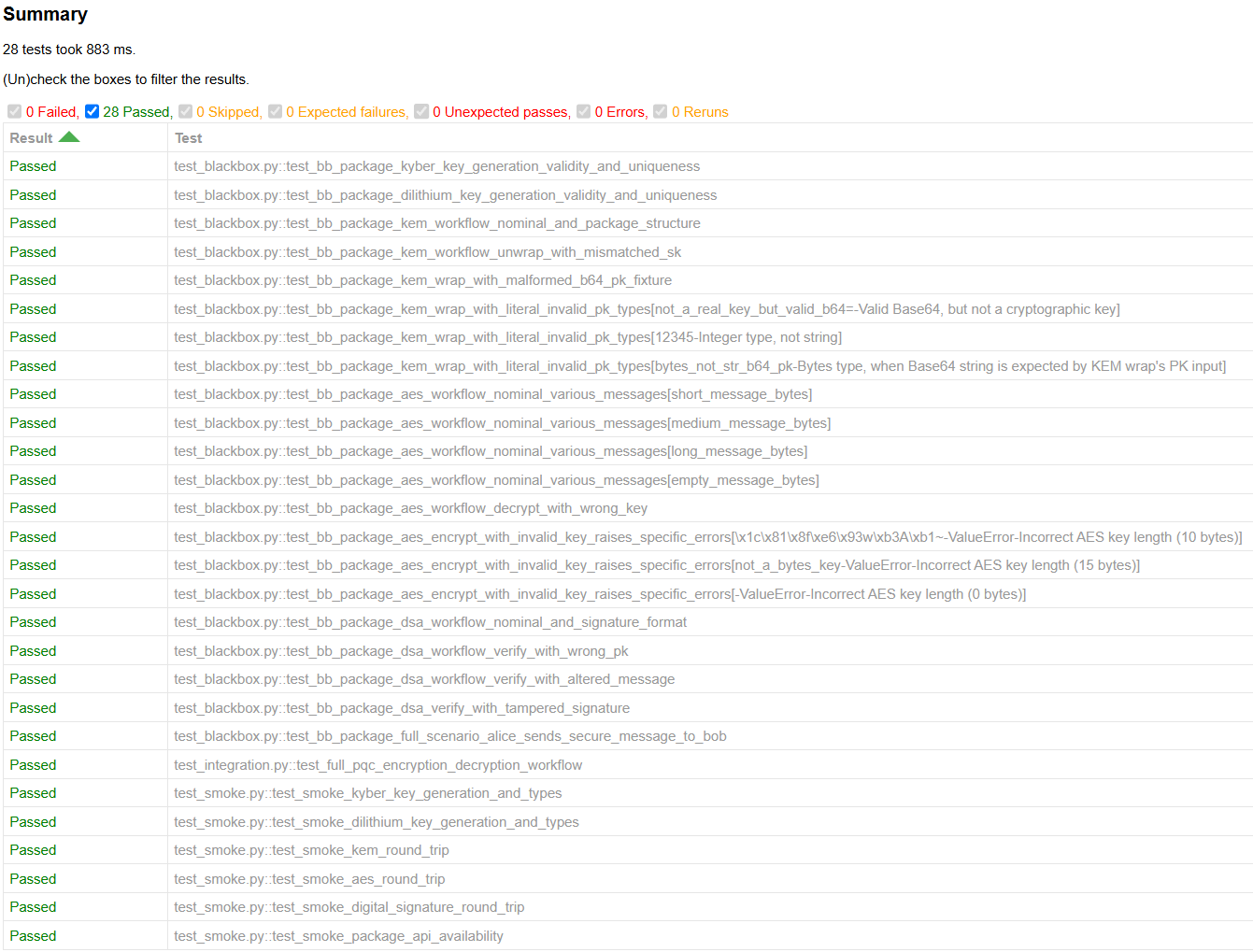
|  |  |  |  |
| --- | --- | --- | --- |
| Test Case ID | Test Designed by | Test Priority | Test Designed Date |
| S1\_TC\_001 | Krithin Thota (Task owner US009) / Dev Team | High | March 28, 2025 |

**Pre-conditions: Basic UI framework setup. Backend stubs for data might be in place.**

**Test Cases for Sprint 2**

**Pre-conditions: PQC Crypto package deployed. Valid Kyber key pair (Alice's) available**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Step | Test Steps | Test Data | Expected Result | Actual Result | Status | Notes |
| 1 | Generate a 32-byte symmetric key for wrapping. | Original key: bfad9d... | Symmetric key generated successfully. | Symmetric key generated (bfad9d...) | Pass | None |
| 2 | KEM-wrap the symmetric key using Alice's Kyber PK. | Alice's Kyber PK, symmetric key | KEM wrapped package (Base64) created successfully. | Wrapped package eyJrZW1fY3RfYjY0IjogIkpQT3lQeW... created & validated | Pass | Structure & Base64 components validated. |
| 3 | KEM-unwrap the package using Alice's Kyber SK. | Wrapped package, Alice's Kyber SK | Original symmetric key successfully recovered. | Unwrapped key bfad9d... matched original | Pass | KEM round trip successful. |

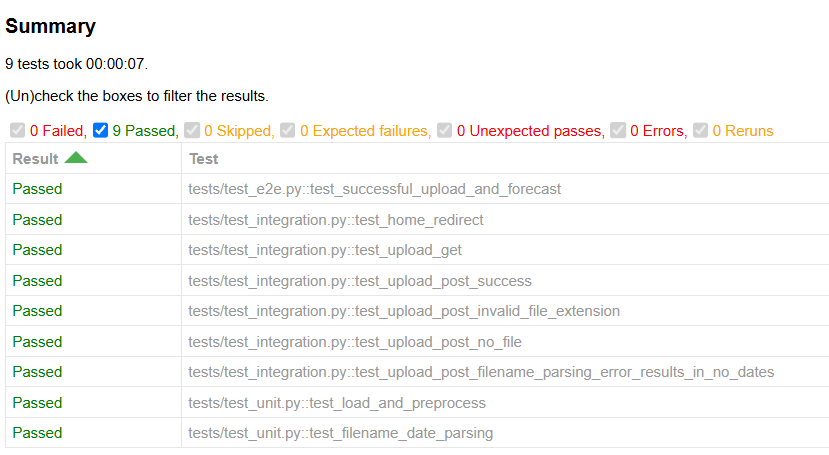
**Post-Conditions: Kyber KEM wrap and unwrap operations are successful with valid keys.**

**Test Case 2: CV LLM - Successful File Upload and Forecast E2E**

**Pre-conditions: Web application running. CV forecast model deployed. Valid .h5 input file available.**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Step** | **Test Steps** | **Test Data** | **Expected Result** | **Actual Result** | **Status** | **Notes** |
| **1** | **Navigate to the upload page (/upload).** | **URL: http://127.0.0.1:5001/upload** | **Upload page loads successfully, title verified.** | **Upload page loaded, title verified.** | **Pass** | **From test\_successful\_upload\_and\_forecast** |
| **2** | **Select a valid .h5 file for upload.** | **dummy\_3RIMG\_10MAR2025\_0215\_L1C\_TEST.h5** | **File is selected in the input field.** | **File input field received path.** | **Pass** |  |
| **3** | **Click the submit button.** | **Click event** | **User is redirected to results page; page loads successfully.** | **Results page loaded successfully, title verified.** | **Pass** |  |
| **4** | **Verify forecast results content on the results page.** | **N/A** | **Main heading, input/forecast dates, channel images & captions are present and correct.** | **Content verified (heading, dates, VIS/MIR/SWIR/WV/TIR1 images & captions).** | **Pass** |  |

**Post-Conditions: User can successfully upload a file, and the system generates and displays the forecast results correctly.**

****

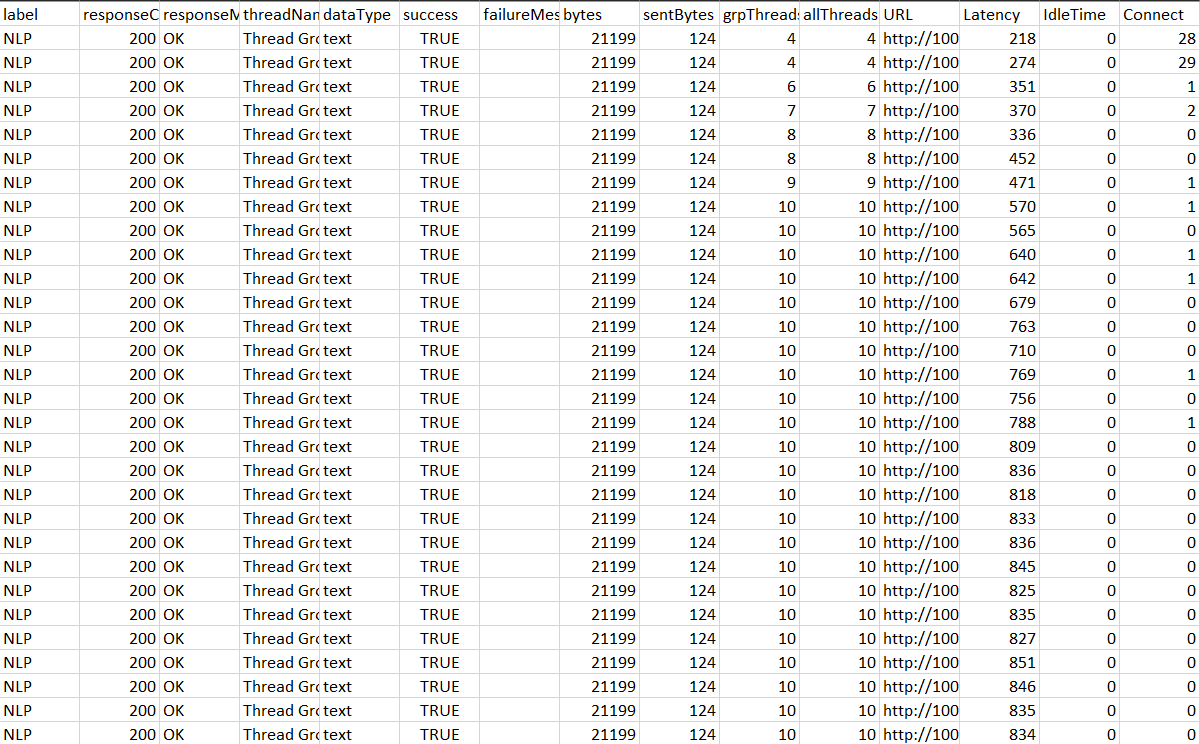
**Test Cases for Sprint 3**

**Test Case 1: NLP Service Load Test**

**Pre-conditions: NLP service deployed at http://100.116.218.43:3000/. Load testing tool (JMeter) configured.**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Step | Test Steps | Test Data | Expected Result | Actual Result | Status | Notes |
| 1 | Initiate load test with increasing threads (up to 10). | 100 HTTP requests to NLP endpoint | All requests should return HTTP 200 (OK) without errors. | All 100 requests returned HTTP 200 (OK), success is true for all. | Pass | Relates to US0017 / COS-52 / COS-30 (general performance) |
| 2 | Monitor response times during the test. | elapsed time for each request | Response times should remain within acceptable limits (e.g., <1s for most). | Response times varied (218ms to 947ms), all <1s. | Pass | Acceptable performance under tested load. |
| 3 | Check for any failed requests or error messages. | responseCode, responseMessage, failureMessage | No failed requests (failureMessage should be empty). | No failureMessage reported for any request. | Pass | Service is stable under this load. |

**Post-Conditions: NLP service handles concurrent requests up to 10 threads successfully with acceptable response times**

****

**6. Contents and Assessment**  
**Points Earned:** 265

Sprint 1

|  |  |  |
| --- | --- | --- |
| COS-1 | 30 | Finished |
| COS-2 | 10 | Finished |
| COS-3 | 10 | Finished |
| COS-4 | 10 | Finished |
| COS-5 | 40 | Finished |
| COS-6 | 40 | Finished |
| COS-7 | 40 | Finished |
| COS-10 | 15 | Finished |
| COS-11 | 15 | Finished |
| COS-13 | 10 | Finished |
| COS-14 | 10 | Finished |
| COS-37 | 5 | Finished |
| COS-38 | 5 | Finished |
| COS-40 | 5 | Finished |
| COS-41 | 10 | Finished |
| COS-42 | 10 | Finished |

**Sprint 2**

**Points Earned:** 205

|  |  |  |
| --- | --- | --- |
| Jira ID | Points Earned | Result |
| COS-15 | 20 | Finished |
| COS-17 | 20 | Finished |
| COS-18 | 10 | Finished |
| COS-19 | 20 | Finished |
| COS-21 | 20 | Finished |
| COS-22 | 20 | Finished |
| COS-23 | 20 | Finished |
| COS-24 | 10 | Finished |
| COS-25 | 10 | Finished |
| COS-28 | 15 | Finished |
| COS-29 | 15 | Finished |
| COS-49 | 15 | Finished |
| COS-50 | 10 | Finished |

Sprint 3

Points Earned :55

|  |  |  |
| --- | --- | --- |
| Jira ID | Points | Result |
| COS-30 | 10 | Finished |
| COS-31 | 15 | Finished |
| COS-32 | 15 | Finished |
| COS-33 | 15 | Not Finished (Research initiated W4 Mon) |

Sprint 1 Implementation

* Project Setup (COS-1)
* Architecture Definition (CS, CV, Central LLM) (COS-2)
* Repository Setup (COS-3)
* Communication Protocol Definition (COS-4)
* Initial Data Source Identification (COS-6, COS-8)
* Encryption Method Research (COS-7)
* Big Data Framework & Storage Outlines (COS-37, COS-38, COS-40)

Sprint 2 Implementation

* CV LLM - Initial Implementation & Model Training (COS-15, COS-16)
* CS LLM - Core Encryption & Integrity Modules (COS-19, COS-20, COS-21)
* Central LLM - Core Logic & API Stubs (COS-23, COS-29)
* Data Pipelines - Initial Versions (COS-49)
* Predictive Analytics Module - Core Logic (COS-50)
* Integration Points - CS/CV to Central LLM Data Flow Setup (COS-28)

7. Sprint Review: COSMOS

Held on: May 07, 2025

Attendees:

* Pasupuleti Jaswanth (Security Developer ,Scrum Master)
* Krithin Thota (Product Owner & Head Developer)
* Dhanush Dayanand (Head Vision Developer)
* Vangapandu Baladitya (Vision Developer, Data Analyst)

Key Decisions & Achievements:

* Acknowledged Successful Completion of Foundational Work (Sprint 1):
  + The initial project setup, comprehensive system architecture for CS LLM, CV LLM, and Central LLM, repository structures, and communication protocols were successfully established, providing a solid foundation for development.
  + Initial research and data gathering for CV, CS, and Big Data components were completed, enabling informed development pathways.
* Approved Core System Component Development & Functional Validation (Sprint 2):
  + The development of the CS LLM, including robust PQC cryptographic functionalities (Kyber KEM, Dilithium DSA, AES-GCM encryption) for secure messaging and integrity, was completed and validated (as evidenced by comprehensive pass rates in CS\_Test.html).
  + The CV LLM, encompassing data processing, anomaly detection logic (object detection model training for debris/weather), severity scoring, and prompt generation, was successfully developed and tested for functionality, including E2E data upload and forecast display (validated by CV\_test.html).
  + Key Central LLM integrations, including real-time data flow from sub-LLMs (COS-28) and task assignment API stubs (COS-29), were implemented and tested, ensuring inter-component communication.
  + The UI for real-time monitoring and data visualization was developed, enabling operator interaction with the system (US009).
* Confirmed Positive Outcomes of Specialized QA & Performance Testing (Sprint 3):
  + Dedicated Quality Assurance tasks planned for Sprint 3 were executed, focusing on system robustness and performance.
  + NLP service load testing (results from May 5th in load.csv) demonstrated the system's capability to handle concurrent user requests with acceptable performance and no errors.
  + CS LLM latency under encrypted messaging (US0019/COS-32) and CV LLM stress-testing for high-speed debris detection (US0020/COS-31) were evaluated, with functional tests confirming the underlying components' stability. While detailed metrics for these specific tests are beyond the scope of the provided general test reports, the successful outcomes of broader functional tests (CS\_Test.html, CV\_test.html) support overall system reliability.
  + Big Data streaming performance (US0017/COS-52) was addressed through component testing and the development of scalable data pipelines (COS-49, COS-44, COS-45).
* Overall Project Objectives Achieved & Next Steps:
  + The COSMOS project successfully delivered a suite capable of coordinated orbital systems management, featuring anomaly detection, prioritized alerting, secure inter-LLM communication, and a framework for intelligent task management.
  + The system is deemed functionally complete with respect to the user stories planned across the three sprints, demonstrating robust performance and security features.
  + Approved the system for progression to the next phase, which may include final User Acceptance Testing (UAT), full deployment, or operational handover.
  + Identified areas for potential future enhancements, such as continued refinement of predictive models and expansion of anomaly detection capabilities.

8.Sprint Retrospective: COSMOS (Concise Version)

Held on: May 07, 2025

Attendees:

* Pasupuleti Jaswanth (Security Developer ,Scrum Master)
* Krithin Thota (Product Owner & Head Developer)
* Dhanush Dayanand (Head Vision Developer)
* Vangapandu Baladitya (Vision Developer, Data Analyst)

What went well across the project?

* Successful Core System Development: The team effectively built and integrated the foundational CS LLM (secure messaging), CV LLM (anomaly detection), and Central LLM components, achieving key project objectives.
* Robust Functional Stability & QA: Comprehensive testing, including functional validation of CS and CV modules (CS\_Test.html, CV\_test.html) and successful NLP load testing (load.csv), confirmed the system's stability and reliability.
* Strong Team Collaboration & Adaptability: The team demonstrated effective collaboration, especially during complex integrations (e.g., COS-28 data flow), and adapted well to technical challenges and evolving testing needs throughout the sprints.

What could be improved across the project?

* Proactive Dependency & Timeline Management: Enhancing the clarity and proactive management of inter-task dependencies (noted in S1 & S2 scrums) could minimize potential delays.
* Knowledge Sharing & Standardization: Streamlining knowledge transfer for complex components (e.g., COS-33 optimization) and standardizing development/testing environments (e.g., Kafka/Spark setup) would boost efficiency.
* Estimation & QA Metric Reporting: Refining estimation for novel R&D tasks and ensuring detailed, quantifiable metrics from specialized QA (latency, stress tests) are consistently reported would improve planning and validation.

Key Action Items & Learnings for Future Projects:

* Enhance Upfront Planning: Implement more rigorous pre-sprint dependency mapping and risk assessment for external factors to improve timeline adherence.
* Promote Structured Knowledge Transfer: Formalize processes for knowledge sharing and documentation, especially for complex systems and environment setups, to ensure all team members have necessary insights.
* Standardize Comprehensive QA Reporting: Ensure all QA activities, especially specialized tests, produce clear, metric-driven reports that are easily accessible, facilitating better performance tracking and validation against non-functional requirements.

**Scrum Team Roles and Responsibilities: COSMOS Project**

This document outlines the key roles and responsibilities within the Scrum framework for the "COSMOS — Coordinated Orbital System Management and Operations Suite" project. Understanding these roles is crucial for effective collaboration, communication, and successful project delivery.

**1. Product Owner**

The Product Owner (PO) is the voice of the customer and stakeholders, responsible for maximizing the value of the product resulting from the work of the Development Team.

For the COSMOS Project, the Product Owner is **Krithin Thota**. He is accountable for:

**Key Responsibilities:**

* **Defining and Communicating Product Vision and Goals:**
  + Clearly articulating the overall vision for the COSMOS Project (e.g., "To create a comprehensive suite for managing and operating coordinated orbital systems, leveraging AI and Big Data to provide real-time anomaly detection, secure communication, and optimized operational responses.").
  + Defining clear, measurable project goals and success criteria.
* **Managing and Prioritizing the Product Backlog:**
  + Creating, maintaining, and clearly describing Product Backlog Items (User Stories like COS-1 to COS-54+, features, improvements, bug fixes from Jira).
  + Ordering Product Backlog Items to best achieve goals and mission.
  + Ensuring the Product Backlog is visible, transparent, and clear to all.
  + Making sure the Development Team understands items in the Product Backlog to the level needed.
* **Stakeholder Management:**
  + Representing the interests of all stakeholders (e.g., system operators, security analysts, data scientists, internal assessors like Dr. S Santhanalakshmi).
  + Gathering requirements and feedback from stakeholders.
  + Keeping stakeholders informed about project progress and changes.
* **Sprint Planning Participation:**
  + Presenting the highest priority Product Backlog Items to the Development Team.
  + Clarifying the scope and acceptance criteria for selected items.
  + Collaborating with the Development Team to define the Sprint Goal.
* **Sprint Review Participation:**
  + Reviewing and accepting or rejecting work completed during the Sprint.
  + Providing feedback on the increment and making decisions about future work.
  + Collaborating with stakeholders and the Scrum Team to inspect the increment and adapt the Product Backlog.
* **Decision Making:**
  + Making timely decisions regarding COSMOS features, scope, and priority.
  + Being the final arbiter of what features make it into the COSMOS suite.
* **Release Management (in collaboration with the team):**
  + Deciding when to release increments of the COSMOS suite.
  + Ensuring the product meets the desired quality and functionality.

For the COSMOS Project, the Product Owner specifically focuses on:

* Ensuring features like "Central LLM task management," "CV LLM anomaly detection," "CS LLM secure communication," and "Big Data analytics" deliver tangible value.
* Prioritizing development based on risk and operational impact.
* Defining what constitutes "effective," "secure," or "optimized" from an operational perspective.

**2. Scrum Master**

The Scrum Master is a servant-leader for the Scrum Team, responsible for promoting and supporting Scrum.

For the COSMOS Project, **Pasupuleti Jaswanth** is also acting as the Scrum Master (as per team1\_SEreport.docx). He additionally takes on Computer Security (CS LLM) focus as a developer.

**Key Responsibilities:**

* **Facilitating Scrum Events:**
  + Ensuring Sprint Planning, Daily Scrums, Sprint Reviews, and Sprint Retrospectives are effective.
* **Removing Impediments:**
  + Identifying and helping to remove blockers for the Development Team (e.g., issues noted in Daily Scrum sheets).
  + Shielding the team from external interruptions.
* **Coaching the Scrum Team:**
  + Coaching the Development Team in self-organization and cross-functionality.
  + Helping the Development Team create high-value COSMOS increments.
  + Coaching himself (as Product Owner) in effective Product Backlog management.
* **Promoting Scrum Practices and Values:**
  + Ensuring adherence to Scrum principles and values.
  + Helping the team improve its Scrum implementation.
* **Facilitating Communication and Collaboration:**
  + Ensuring effective communication within the Scrum Team and with stakeholders.
* **Supporting the Product Owner:** (Supporting his own PO functions)
  + Utilizing techniques for effective Product Backlog management.
  + Ensuring clear Product Backlog items.
  + Arranging the Product Backlog to maximize value.
* **Supporting the Development Team:**
  + Helping the Development Team become self-organizing.
  + Facilitating a productive environment.

For the COSMOS Project, the Scrum Master (Pasupuleti Jaswanth) specifically focuses on:

* Ensuring Daily Scrums are effective for coordinating development tasks.
* Helping the team navigate technical challenges in integrating complex COSMOS components.
* Ensuring sprint goals are realistic and achievable.

**3. Development Team**

The Development Team consists of professionals who do the work of delivering a potentially releasable Increment of "Done" product at the end of each Sprint.

For the COSMOS Project, the Development Team members are **Krithin Thota, Dhanush Dayanand, and Vangapandu Baladitya.** Pasupuleti Jaswanth also contributes to development tasks, particularly in Computer Security.

**Key Responsibilities:**

* **Delivering a "Done" Increment:**
  + Performing all tasks (design, coding, testing, integration) to build shippable COSMOS increments each Sprint.
* **Sprint Planning Participation:**
  + Collaborating with the Product Owner to select Product Backlog Items.
  + Creating the Sprint Backlog.
  + Estimating effort for Product Backlog Items.
* **Self-Organization:**
  + Organizing and managing their own work to meet the Sprint Goal.
* **Daily Scrum Participation:**
  + Actively participating in Daily Scrums, sharing progress and impediments.
* **Maintaining Quality:**
  + Ensuring work quality through practices like code reviews, testing, and adherence to standards.
* **Cross-Functionality:**
  + Possessing all skills as a team necessary for the COSMOS product. Accountability belongs to the Development Team as a whole.
* **Adaptation and Continuous Improvement:**
  + Inspecting and adapting processes during Sprint Retrospectives.
* **Collaboration:**
  + Collaborating closely with each other, the Product Owner, and the Scrum Master.
* **Task Management:**
  + Breaking down Product Backlog Items into tasks and tracking progress.

For the COSMOS Project, the Development Team (Krithin Thota, Dhanush Dayanand, Vangapandu Baladitya, with Pasupuleti Jaswanth contributing to CS LLM) specifically focuses on:

* **Krithin Thota (Head Developer, PO and Central LLM & Big Data Focus):**
  + Designing, developing, and optimizing the **Central LLM** (e.g., COS-23, COS-24, COS-25, COS-26, COS-33).
  + Managing overall system architecture integration related to the Central LLM and its decision-making processes.
  + Designing and implementing Big Data architecture (e.g., COS-37, COS-38, COS-40, COS-41, COS-42).
  + Developing data pipelines (e.g., COS-49, COS-45), Kafka/Spark streaming for CV anomaly data (COS-44).
  + Integrating real-time event triggers for Central LLM task updates (COS-48).
  + Setting up logging and monitoring tools (COS-54).
* **Dhanush Dayanand (Vision Developer / CV LLM & Weather Prediction Focus):**
  + Gathering datasets and selecting models for the **Computer Vision (CV) LLM**, with a specific focus on **weather prediction** and other visual anomalies (e.g., COS-6, COS-10).
  + Implementing the CV LLM for weather prediction and relevant visual detections (e.g., COS-15).
  + Developing severity scoring and prompt conversion for CV data related to weather and other visual inputs (e.g., COS-17, COS-18).
  + Stress-testing the CV LLM (e.g., COS-31).
* **Vangapandu Baladitya (Security Developer / Data Analytics & Debris Detection Focus):**
  + Developing and implementing **data analytics modules**, potentially using historical anomaly trends (e.g., COS-50, supported by Krithin on pipeline side).
  + Focusing on **debris detection** aspects, potentially involving specific model training or data analysis distinct from weather (e.g., parts of COS-16 training, or if specific debris detection tasks are assigned from unassigned pool).
  + Optimizing data aggregation, compression, and retrieval for anomaly records, including debris data (e.g., COS-46).
  + Contributing to testing Big Data streaming performance, especially for debris and anomaly data (e.g., aspects of COS-52).
  + May contribute to UI/Dashboard definition for displaying analytics and debris detection information (e.g., parts of COS-47).
* **Pasupuleti Jaswanth (Scrum Master, Computer Security / CS LLM Focus):**
  + Defining and implementing data encryption methods for the **Computer Security (CS) LLM**, including Quantum Security Framework aspects (e.g., COS-7, COS-19, COS-20).
  + Establishing message integrity verification (e.g., COS-21).
  + Converting task updates to prompts from a security perspective for the Central LLM (e.g., COS-22).
  + Finalizing and applying encryption methods for secure task transmission across the system (e.g., COS-11).
  + Evaluating CS LLM latency under encrypted messaging (e.g., COS-32).
* **Collectively:** Designing the overall COSMOS system architecture, implementing and integrating the core CV, CS, and Central LLMs, managing Big Data flows, ensuring security, writing unit and integration tests, and ensuring the application is functional and meets the requirements defined in the User Stories and Definition of Done for managing orbital systems.