

SOFTWARE RELEASE PLAN

CitiSense

Revision History

Version	Date	Description	Author
1.0	05/25/2025	Initial Draft	QuadThink
1.1	05/29/2025	Revised Template Draft	QuadThink

Table of Contents

1	INTRODUCTION	4
1.1	PURPOSE OF THE RELEASE PLAN	4
1.2	GOALS AND OBJECTIVES OF THE RELEASE:	4
1.3	SCOPE OF THE RELEASE (WHAT'S INCLUDED AND EXCLUDED):.....	4
1.3.1	INCLUDED:	4
1.3.2	EXCLUDED:.....	5
2	RELEASE INFORMATION	5
3	TEAM AND ROLES.....	5
3.1	KEY STAKEHOLDERS:.....	5
3.1.1	HUMAN RESOURCES	5
3.1.2	DEVELOPMENT TEAM:	5
3.1.3	TESTING TEAM:	5
3.1.4	RELEASE MANAGEMENT:	6
3.1.5	OTHER RELEVANT ROLES:.....	6
4	SCOPE AND FEATURES.....	6
4.1	LIST OF FEATURES TO BE INCLUDED IN THE RELEASE:	6
4.2	USER STORIES OR REQUIREMENTS:	6
4.3	DEPENDENCIES:	6
4.4	OUT-OF-SCOPE ITEMS:	7
5	RELEASE SCHEDULE	7
5.1	TIMELINE OF ACTIVITIES:	7
5.2	KEY MILESTONES:	8
6.	TESTING STRATEGY	8
7.	DEPLOYMENT PLAN.....	11
8.	COMMUNICATION PLAN.....	13

9. RISK MANAGEMENT	13
10. ROLLBACK PLAN	14
11. POST-RELEASE ACTIVITIES.....	15

1 Introduction

1.1 Purpose of the release plan

To outline the strategy for releasing the Sentiment Analysis Web App, an AI automated sentiment analysis that processes data in Filipino and English, provides data visualization features, and displays insights for client services.

1.2 Goals and objectives of the release:

- Develop an automated sentiment detection system that will help users efficiently analyze a large volume of user feedback.
- Support multilingual sentiment detection (English and Tagalog) and provide a visualization dashboard feature with filters for easier data interpretation.
- Reduce the manual analysis time by 70% through automation and visualization representation of sentiment results.
- Support the government institution through the development and deployment of an AI-powered system that streamlines the analysis of user feedback.

1.3 Scope of the release (what's included and excluded):

1.3.1 Included:

- User Authentication
- Fine-tuned NLP multilingual sentiment analysis model (Tagalog and English)
- Visualization dashboard with filters
- Downloadable reports in .png or .pdf format

1.3.2 Excluded:

- Data Import feature
- Customizable dashboards (future implementation)
- Further fine-tuning sentiment analysis model (future implementation)

2 Release Information

- Release name & version: CitiSense v1.0
- Target release date: December 13, 2025
- Release type: Release (major)

3 Team and Roles

3.1 Key stakeholders:

- QuadThink Team
- DOST—STII Information Resources and Analysis Division (IRAD)

3.1.1 Human Resources

- Advisor: Manuel Calimlim Jr.
- Technical Support: IT Department of DOST-STII

3.1.2 Development team:

- Project Manager: Lorenzo Emil Bernal
- Lead Developer: Lorenzo Emil Bernal
- Frontend Developers: Gabrielle Cabangcala, Luis Lorenzo Lazaro
- Backend Developer: Lorenzo Emil Bernal, Mark Lyster Marabi

3.1.3 Testing team:

- QA Lead: Manuel Calimlim Jr., Mark Lyster Marabi
- Testers: Lorenzo Emil Bernal, Luis Lorenzo Lazaro, Gabrielle Cabangcala

3.1.4 Release Management:

- Release Manager: Mark Lyster Marabi
- Release Team: Lorenzo Emil Bernal, Luis Lorenzo Lazaro, Gabrielle Cabangcala

3.1.5 Other relevant roles:

- Compliance Advisors: Legal and Compliance Team

4 Scope and Features

4.1 List of features to be included in the release:

- Authentication: Allow DOST-STII's IRAD department to register and login with their account to the system.
- Dashboard View: Allow the user to view dashboard after logging in.
- Specific-Service Analysis: Allow users to select individual services to analyze visualizations.
- Overall Analysis: Provide analysis and visualization to all services
- Visualization: Provides charts, and sentiment trends for the services.
- Data Exportation: Enables the user to export the visualized data into .png, or pdf.
- View past visualizations: Allows the user to access previous data visualization.

4.2 User stories or requirements:

- As an IRAD Department personnel, I can use data visualizations and download details or older information for effective reporting as an employee of DOST STII.
- As an employee of DOST, I have access to the sentiment analysis platform to view and work with feedback from the DOST-STII services.

4.3 Dependencies:

- Setting up the user database has been completed, for users registering or accessing their DOST accounts.
- Fine-tune and deploy an NLP model so that it can handle text processing and sentiment analysis.
- Enable the system to store past visualization data upon exporting visualization data.

4.4 Out-of-scope items:

- File upload

5 Release Schedule

5.1 Timeline of activities:

Start Date	End Date	Phase	Activities
4-1-2025	5-31-2025	Q1 – Requirements Gathering	<ul style="list-style-type: none"> • Weekly stakeholder interviews and update, • Define core features • Create and finish documentations • Create initial use case and wireframes
6-13-2025	10-31-2025	Q2 – Development of the CitiSense web application	<ul style="list-style-type: none"> • Complete the UI/UI design for the CitiSense web application • fine-tune selected NLP model for core functionality • Develop core features, and backend/frontend integration • Provide weekly progress updates to stakeholders, and • Conduct monthly sprint reviews to gather feedback from stakeholders and adjust priorities.
11-1-2025	12-4-2025	Q3 – Testing and Deployment	<ul style="list-style-type: none"> • Host the frontend and backend through hosting platforms • Execute Unit Testing, Integration Testing, and System Testing done by the QA team
12-5-2025	12-14-2025	Q3 – Testing and Deployment	<ul style="list-style-type: none"> • Execute User Acceptance Testing (UAT) with the stakeholders (DOST-STII IRAD department). • Initiate code freeze after UAT.
12-15-2025	1-15-2025	Q3 – Testing and Deployment	<ul style="list-style-type: none"> • Configure any requirements • Deploy the CitiSense web application through a production environment, • Initiate release sign-off after.
2-1-2025	06-01-2025	Q4 – Post-Deployment Maintenance	<ul style="list-style-type: none"> • Start providing bug fixes that occur after deployment • Further model fine-tuning if needed.

--	--	--	--

5.2 Key milestones:

- Development completion: December 13, 2025
- Testing phases:
 - Unit Testing Completion: November 13, 2025
 - Integration Testing Completion: November 27, 2025
 - System Testing Completion: December 4, 2025
 - UAT Completion: December 13, 2025
- Code freeze Period: December 14, 2025
- Deployment Date: January 6- 15, 2026
- Release sign-off: January 15, 2026
- Dependencies between tasks:
 - All software features must be developed before testing can begin
 - All testing phases must be complete before deployment can begin

6. Testing Strategy

- Types of testing to be performed:
 - Unit Testing – Developers will individually test modules such as authentication, sentiment processing, data visualization, and data export functions to verify that each part functions correctly.
 - Integration Testing – Ensures that modules like the NLP model, visualization dashboard, and authentication system interact correctly with one another.
 - System Testing – End-to-end testing of the full web application to confirm that all components work as expected in a complete, user-ready system.

- User Acceptance Testing (UAT) – Conducted with stakeholders from DOST STII-IRAD to confirm that the application meets user functional and usability expectations.
- Testing environment:
 - Development Environment: Used by QuadThink developers to build and unit-test features such as sentiment analysis, authentication, and data visualization. This environment runs locally or on isolated cloud containers to allow safe iteration.
 - QA (Quality Assurance) Environment: Used for integration and system testing. After unit testing, features are pushed here to test interactions between backend APIs (Django), the frontend (Vercel), and the MySQL database. Issues found here are logged and resolved before proceeding.
 - Staging Environment (for UAT): A near-identical replica of the production environment where User Acceptance Testing (UAT) is conducted by DOST STII-IRAD personnel. All features, data flows, and exports are tested with realistic conditions and sample feedback to simulate actual use.
 - Production Environment: The live system accessed by official users once the app is deployed. Hosted via Vercel (frontend) and Google Cloud Run or App Engine (backend), with MySQL handling live data. This environment will be used during the official launch phase after UAT sign-off.
- Test data:
 - Mock English/Filipino feedback with varied sentiment.
 - Edge cases/Extreme or unique inputs (e.g., empty, long, mixed-language entries).
 - Sanitized DOST feedback during UAT (if available).
- Entry and exit criteria for each testing phase: (Examples)
 - Unit Testing
 - Entry Criteria: Code for the module or feature is complete.
 - Exit Criteria: 100% of unit test cases pass with no major logic errors.
 - Integration Testing

- Entry Criteria: All relevant modules have passed unit testing and are connected.
 - Exit Criteria: Core data flow between frontend, backend, and database works correctly; no unresolved critical issues.
- System Testing
 - Entry Criteria: All modules are integrated and feature complete.
 - Exit Criteria: 95% of system test cases pass; no critical or high-priority bugs remain.
- User Acceptance Testing (UAT)
 - Entry Criteria: System is stable and deployed to the staging environment.
 - Exit Criteria: Feedback from DOST-STII users is implemented or acknowledged; UAT sign-off received.
- Test automation strategy:
 - Backend tests using unit test or pytest for sentiment analysis model and computational logic.
 - Manual UI testing for dashboards.
 - Vercel logs are used for live monitoring after deployment.
- Defect management process:
 - Use GitHub Issues or Google Sheets for tracking.
 - Prioritize by severity (Critical, Major, Minor).
 - Assign fixes, verify in QA or Staging.
 - Weekly bug reports shared with client during UAT.
- Severity Levels:
 - **Critical:**
 - authentication failure
 - broken sentiment analysis processing
 - complete application crash
 - **High:**
 - data exports
 - real-time feedback processing
 - survey response handling
 - **Medium:**

- inaccurate sentiment labels
 - delayed data and visualization display
 - minor API inconsistencies
- **Low:**
 - layout misalignment/s
 - font inconsistencies
 - color scheme mismatches

7. Deployment Plan

Deployment Approach

- Phased deployments, users from DOST-STII will try the app for pilot testing and when the features are confirmed to be stable, more people within DOST-STII IRAD can join in.

Deployment Environment:

- Vercel, for handling the frontend and serverless tasks
- Django backend hosting (e.g., Render, AWS)
- MySQL, for the main database.

Deployment Steps

- Prepare Production Environment
- Configure Application & Services
- Conduct Final Testing in Production Environment
- Pilot test to Internal DOST-STII teams to monitor performance and address any issues

Rollback Plan

- On Vercel, the web app allows you to revert instantly to your latest successful production deployment.
- Backend (Django): Use tools from the hosting site to roll back to a previous version or save previous versions of backend source code and revert if needed.
- Reverting updates made to the schema in Database (MySQL) can be done using migrations or data restoration can be done using backups.

Rollback Conditions:

Rollback actions are taken when:

- A critical system takes place when the production web application cannot be accessed, errors are returned from the server side, or the backend API fails to react or act strangely after changes are made.
- If data becomes corrupted, lost, or incorrect records in the database after deployment, this can affect the integrity of the data.
- Problems may occur like, a key feature that stops working, important user processes cannot be used, important security flaws are found that could make some information vulnerable.

Rollback Procedure:

- Go to Vercel's dashboard, view your deployment history, and pick out the last version that worked properly. You can use Rollback to bring back the working version and keep the frontend stable again.
- For Django and MySQL, roll back the backend and database to their previous stable version and then re-deploy the code through your server (e.g., Railway or Heroku). If the problem is because of schema migration, either use Django's migrate command or restore the database from your latest backup using mysqldump or the restore tool provided by your host.
- Following rollback, run through the entire application to check if all functions are working correctly. Raise a ticket describing the issue, what caused it, the rollback procedure and how to prevent another occurrence. Alert all relevant organization members, including management.

Estimated Downtime

- Frontend changes should cause little interruptions of frontend is hosted through Vercel without errors. There may be brief outages (under 15 minutes) during a Django (Backend) deployment. Moving databases (up to 30 minutes) may interrupt your service.

Communication Plan for Deployment and Rollback:

- This plan will tell key stakeholders and pilot users when deployments or company maintenance is going to take place. Let employees know about unexpected disruptions through the company's internal system.

8. Communication Plan

- Communication channels:
 - Email
 - Viber
 - Microsoft Teams
 - Web application notifications
- Frequency of communication:
 - Weekly updates during development.
 - Pre-launch and post-launch announcements
 - Weekly status reports to the Information Resources and Analysis Division (IRAD) department.
- Target audience (internal and external):
 - Internal: IRAD department of DOST-STII, developers, data analysts
 - External: DOST clients, service users, investors
- Stakeholder updates:
 - Regular meetings with the project manager, adviser, and developers.
 - Demonstrations of the application to the DOST-STII - IRAD department.

9. Risk Management

Identification of Potential Risks:

Data privacy concerns.

- Data privacy concerns
- Technical issues during deployment
- Security Vulnerabilities

Risk Assessment:

Potential Risk	Probability	Impact	Migration Strategy
Data Privacy Concerns	High	High	Comply with data privacy regulations and delete user data from the database after a set period of time.
Technical Issues during deployment	Medium	High	Adopt test-driven development strategies to reduce bugs; implement rollback strategies.
Security Vulnerabilities	High	High	Secure API endpoints and user input forms

Contingency Plans

- In case of poor model accuracy, the team can fine-tune the NLP model further, used in the CitiSense web application.
- Keep previous stable builds of the CitiSense web application ready for redeployment.

10. Rollback Plan

Conditions for Rollback:

A rollback will be initiated if any of the following critical issues occur during or after deployment:

- Major bugs or system crashes that will affect sentiment classification or dashboard visualization functionalities.
- Inaccurate sentiment analysis results that may cause misinterpretation of user feedback.
- Visualization dashboard unresponsive to new data

Step-by-Step Rollback Procedure:

- Developers consult with the Project Manager, IRAD, and IT teams to assess rollback issues.
- Ensure all logs and feedback data are backed up from the system for analysis.
- Identify the root cause of the problem and propose solutions.

Prepared By: QuadThink

Date: 05/29/2025

Responsibilities:

- **Development Team (QuadThink):** Responsible for executing the rollback and conducting testing after settling the rollback.
- **Infrastructure Support Team (DOST-STII IT Department):** Responsible for supporting the infrastructure of the rollback and ensuring its system stability.
- **Project Management Team:** Responsible for leading the rollback decision-making, coordinating communication within all teams, and making sure that the steps are followed to combat the rollback.
- **Testing Team (QA Lead & Testers):** Responsible for the validation of data after the rollback procedures.

Communication Plan for Rollback

- Regular updates with the DOST-STII teams every day via email, Viber, or MS Teams.
- Short meetings of daily stand-ups during the rollback period to align all teams in addressing the rollback and updating the progress related to it.

11. Post-Release Activities

Monitoring and Support

- Continuous monitoring of the web application.
- Notify users in advance of any maintenance updates.
- Track user interactions to identify and detect any unusual patterns.

Performance Monitoring

- Track key metrics such as error rate of sentiment analysis labels (Positive, Neutral, Negative), to ensure model accuracy and reliability.

Bug Fixes and Hotfixes

- Define a process for identifying, reporting, and resolving reported bugs.
- Determine criteria for urgent hotfixes vs. scheduled patches.

User Feedback Collection

- Directly contact and collect feedback from IRAD users of the web app, through feedback forms.

- Evaluate feedback to improve and enhance the overall user experience.

Release Review and Lessons Learned

- Conduct a formal review of the web application with all the stakeholders to evaluate the success of the release process.
- Document all milestones, challenges, and areas for improvement to be used as a reference for future product updates.