

# Agile Documentation *for* Real-Time AI Sales Call Assistant for Enhanced Conversation Strategies

# TABLE OF CONTENTS

1. Project Overview.....	3
1.1 Project Name.....	3
1.2 Description.....	3
1.3 Goals & Objectives.....	3
1.4 Tech Stack.....	3
2. User Stories & Tasks.....	4
3. Architecture & Design Notes.....	4
3.1 High-Level Architecture.....	4
3.2 Database Schema (Google Sheets).....	6
4. Project Methodologies.....	6
5. Storyboards and Timeline of Project Development.....	6
5.1 Sprint Logs.....	6
5.2 Change Log.....	8
6. Code Documentation.....	8
6.1 Setup Instructions.....	8
6.2 Environment Variables (.env File).....	8
6.3 Running the Project.....	8
7. Demonstration of Project Development.....	9
8. Testing & Deployment Notes.....	10
8.1 Test Cases.....	10
8.2 Deployment Instructions.....	10
9. Final Summary.....	10
9.1 What Went Well:.....	10
9.2 Challenges Faced:.....	11
9.3 Next Steps (Future Enhancements).....	11

# 1. Project Overview

## 1.1 Project Name

Real-Time AI Sales Call Assistant for Enhanced Conversation

## 1.2 Description

The Real-Time AI Sales Call Assistant enhances sales conversations by providing real-time analysis, sentiment detection, product recommendations, and objection handling. It leverages speech recognition, sentiment analysis, and machine learning to transcribe and analyze live sales calls, providing immediate feedback and actionable insights for sales representatives.

## 1.3 Goals & Objectives

- Improve the efficiency and effectiveness of sales representatives.
- Provide real-time feedback and actionable insights during sales calls.
- Enhance customer experience through personalized interactions.
- Enable sales managers to monitor and analyze sales performance.

## 1.4 Tech Stack

Component	Technology Used
Languages	Python
Frameworks	Streamlit
APIs	Google Sheets API, Hugging Face API
Libraries	SpeechRecognition, Vosk, Sentence Transformers, Faiss, Transformers, Pandas, Plotly, PyAudio
Databases	Google Sheets
Other Tools	dotenv (for environment variables)

## 2. User Stories & Tasks

### 2.1 User Stories and Acceptance Criteria

ID	As a	I want to	So that I can	Acceptance Criteria
1	Sales Representative	Receive real-time transcription of my calls	Focus on the conversation	Real-time transcription is accurate and displayed in the application interface
2	Sales Representative	Know the sentiment of the customer's statements	Understand their feelings	Sentiment analysis is performed on transcribed text and displayed with corresponding scores
3	Sales Representative	Receive product recommendations based on the conversation	Suggest relevant products to the customer	Product recommendations are generated and displayed during the call
4	Sales Representative	Handle customer objections effectively	Address their concerns and close the sale	Objection-handling suggestions are provided based on the conversation context

### 2.2 Stakeholder Analysis and Identification

- Sales Representatives: Primary users who utilize the transcription feature during sales calls to improve productivity.
- Sales Managers: Monitor sales performance through call summaries and sentiment analysis, providing feedback for training purposes.
- Development Team: Responsible for building and refining the AI-driven transcription tool, ensuring seamless integration and functionality.
- Customers: Indirect stakeholders who will benefit from improved sales interactions.

## 3. Architecture & Design Notes

### 3.1 High-Level Architecture

- Speech Recognition: Vosk is used to transcribe live audio into text.
- Sentiment Analysis: Utilizes Hugging Face API with a multilingual sentiment model.
- Product Recommendations: Uses Sentence Transformers and Faiss for real-time suggestions.
- Objection Handling: Provides automated responses using trained embeddings.
- Data Storage: Stores and retrieves conversation data from Google Sheets.
- Dashboard: Built with Streamlit to display summaries and sentiment insights.

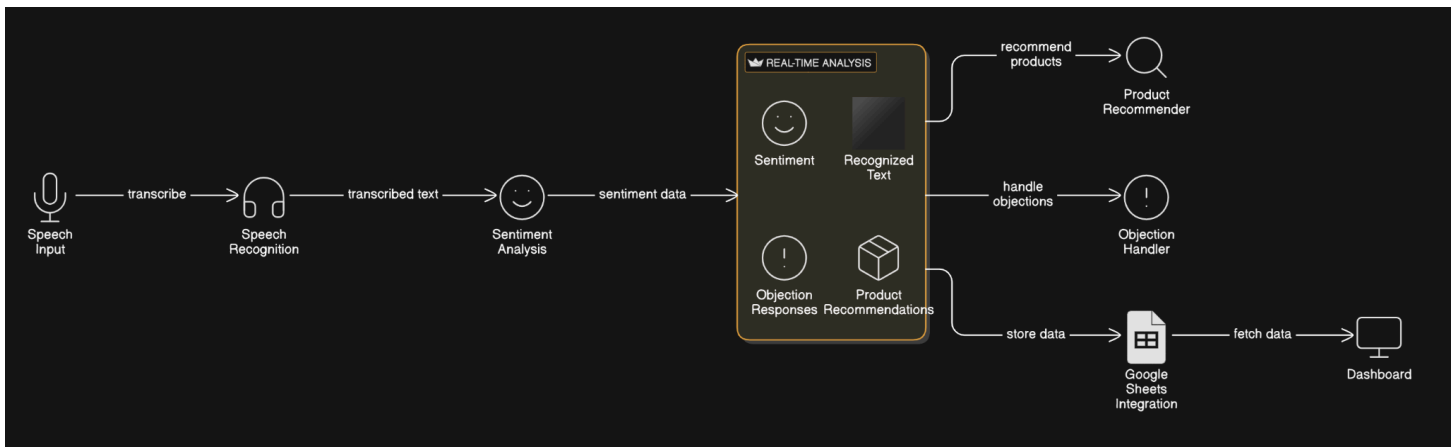
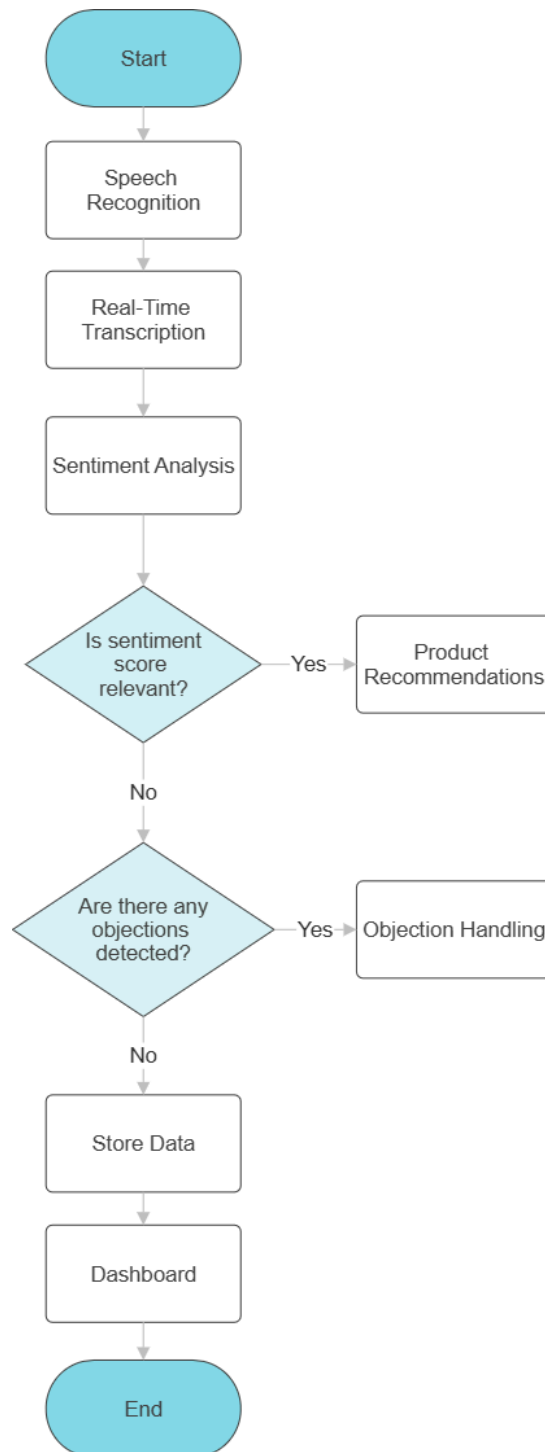


Fig 1. Architecture Diagram (above) and Flowchart (below)



### 3.2 Database Schema (Google Sheets)

Column Name	Description
Call ID	Unique identifier for each sales call
Chunk	A segment of the conversation
Sentiment	Sentiment score for each chunk (0-1)
Summary	Generated summary of the call
Overall Sentiment	Overall sentiment classification (Positive, Neutral, Negative)

## 4. Project Methodologies

### 4.1 Agile Methodology

It was discovered that in previous years, there were traditional software development methods such as waterfall methodology that had many issues and weren't very reliable for effective software and application development. This led to the development of Agile methodology which mainly focuses on agile development. Agile methodology is an iterative approach to project management and software development that focuses on delivering value to customers through continuous improvement, collaboration, and adaptability. While Agile itself is a mindset and set of principles, its implementation often involves specific frameworks like Scrum, Kanban, or Extreme Programming (XP). Here is how we planned to implement our milestones in this project-

- **Sprint Planning:** Define and prioritize user stories, breaking them into manageable tasks.
- **Daily Standups:** Conduct brief meetings to discuss progress, roadblocks, and next steps.
- **Sprint Reviews:** Showcase completed features to stakeholders for feedback.
- **Sprint Retrospectives:** Evaluate sprint performance, identifying improvements for the next cycle.

## 5. Storyboards and Timeline of Project Development

### 5.1 Sprint Logs

- Sprint 1: Set up the project environment and initial configurations.
- Sprint 2: Implement real-time transcription and sentiment analysis.
- Sprint 3: Develop product recommendation and objection-handling features.

- Sprint 4: Integrate features and perform testing.

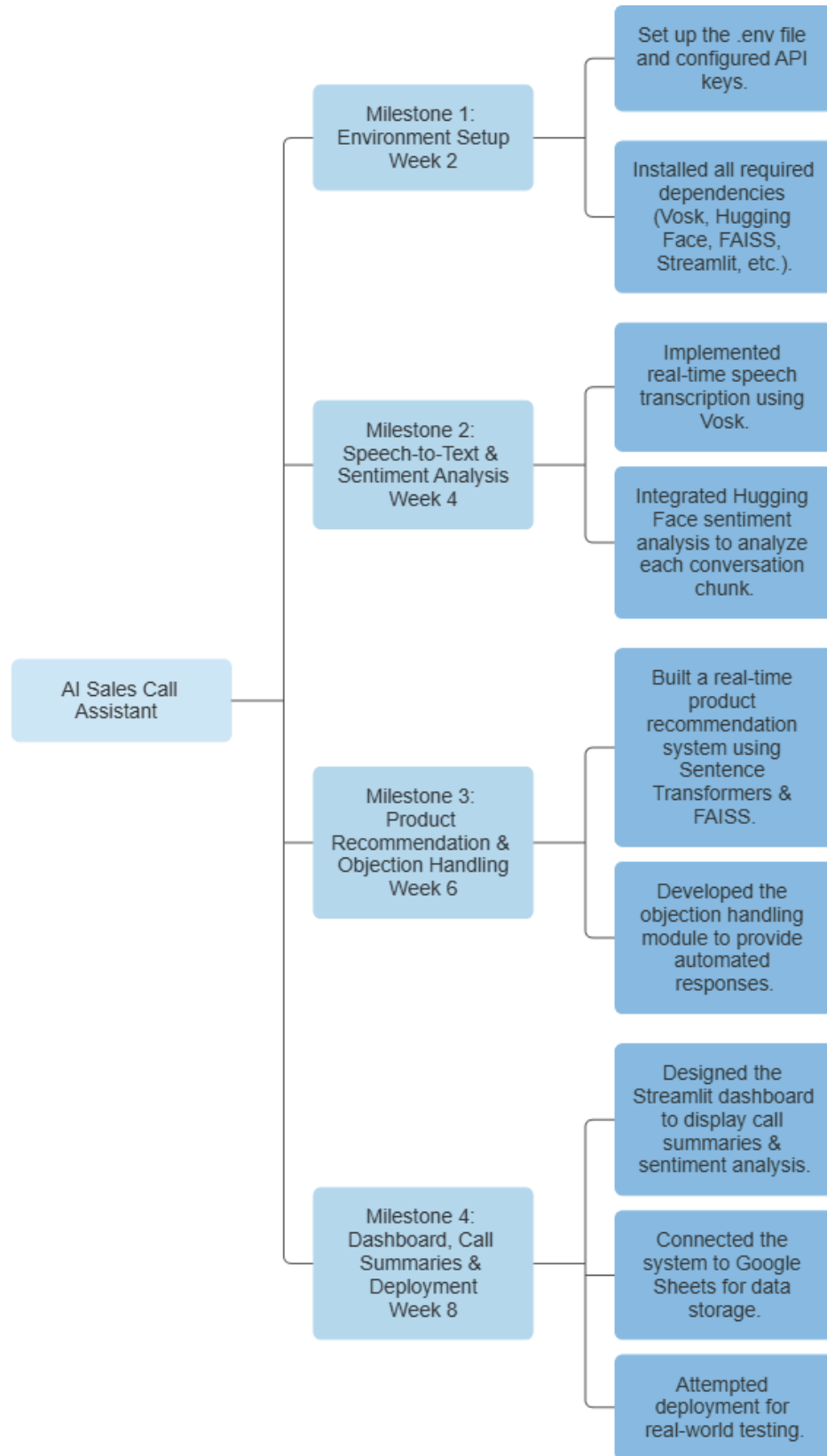


Figure 2. Sprint Map

## 5.2 Change Log

Version	Updates
v0.1	Initial setup, basic transcription.
v0.2	Added sentiment analysis.
v0.3	Implemented objection handling.
v0.4	Added product recommendations.
v1.0	Finalized real-time analysis and dashboard.

## 6. Code Documentation

### 6.1 Setup Instructions

```
# Clone the repository
git clone https://github.com/zia9571/AI-Sales-Call-Assistant.git
cd AI-Sales-Call-Assistant
```

```
# Create a virtual environment
python3 -m venv venv
source venv/bin/activate # On Windows: venv\Scripts\activate
```

```
# Install dependencies
pip install -r requirements.txt
```

### 6.2 Environment Variables (.env File)

```
vosk_model_path=path/to/vosk/model
huggingface_api_key=your_huggingface_api_key
google_creds=path/to/your/google/credentials.json
google_sheet_id=your_google_sheet_id
```

### 6.3 Running the Project

```
streamlit run app.py
```



## 7. Demonstration of Project Development

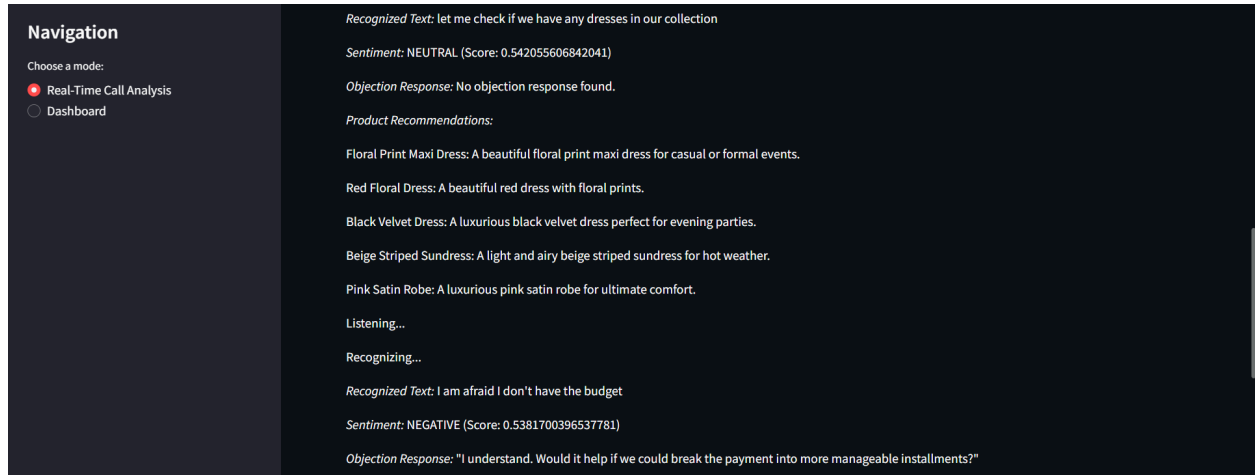


Fig 3. Real-time call analysis

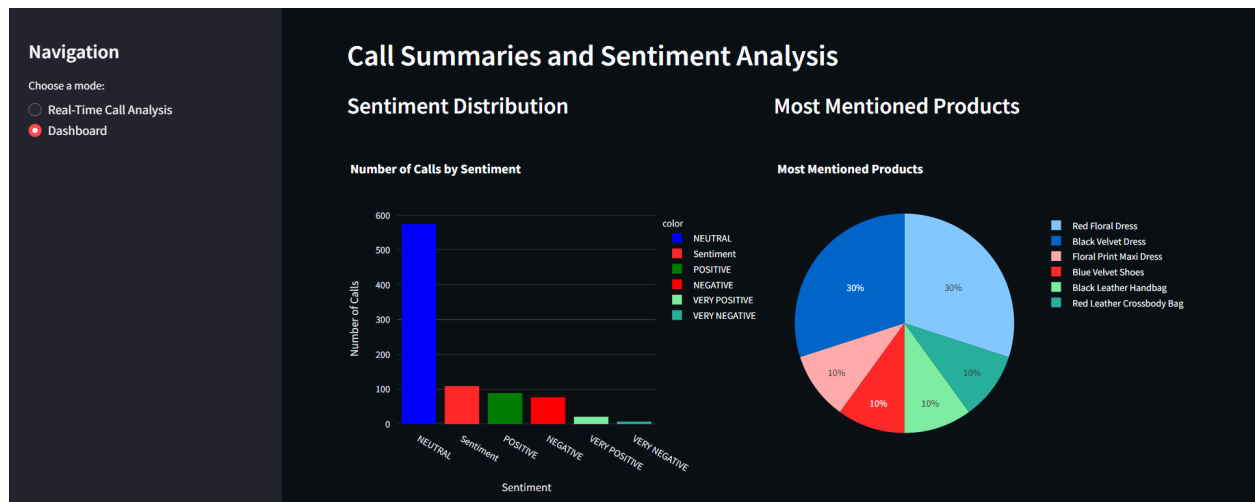


Fig 4. Dashboard containing call insights

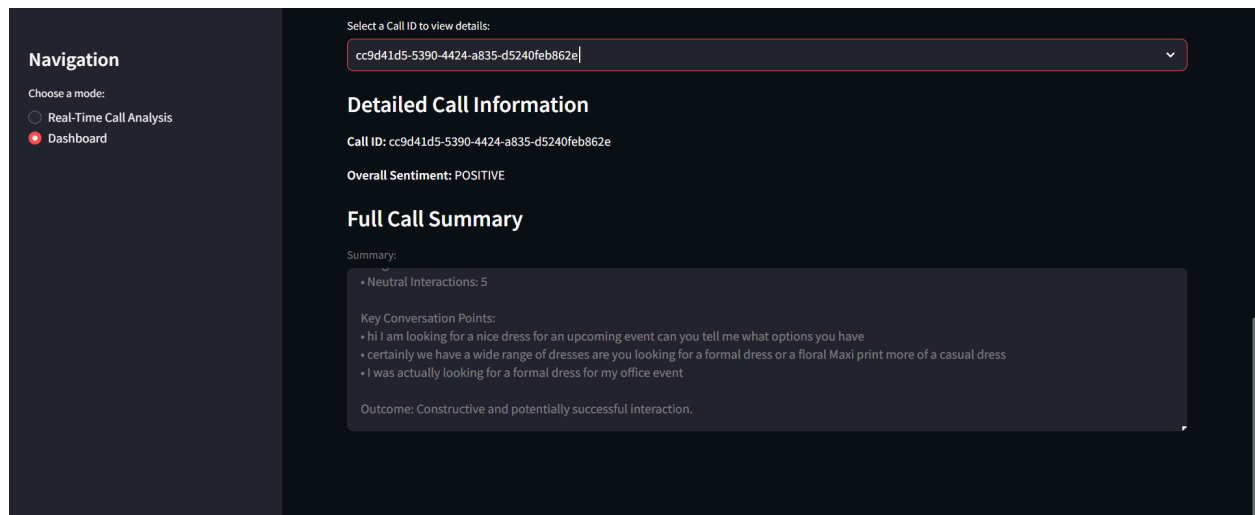


Fig 5. Dashboard containing call summary

## 8. Testing & Deployment Notes

### 8.1 Test Cases

Test Case ID	Test Scenario	Expected Result	Actual Result	Status
TC-01	Transcription accuracy in normal conditions	Transcribed text matches spoken words accurately	Transcription accurate	Passed
TC-02	Transcription accuracy in noisy environment	Minimal errors despite background noise	Minimal errors observed	Passed
TC-03	Sentiment analysis correctness	Sentiment scores align with actual emotions	Sentiment correctly classified	Passed
TC-04	Product recommendation relevance	Suggested products match the conversation context	Relevant products displayed	Passed
TC-05	Voice command functionality	Commands "start listening" and "stop listening" work as expected	Voice commands function correctly	Passed
TC-06	System latency	Transcription delay remains under 1 second	Latency below 1 second	Passed

### 8.2 Deployment Instructions

8.2.1 Local Deployment:

```
python main.py
```

8.2.2 Streamlit Cloud Deployment:

```
streamlit run app.py
```

## 9. Final Summary

### 9.1 What Went Well:

- Successful integration of multiple AI models & APIs.

- Positive feedback from initial users.
- Efficient data storage and retrieval using Google Sheets.

## 9.2 Challenges Faced:

- Handling noisy environments for speech recognition.
- Fine-tuning AI models for better accuracy.
- Managing dependencies and environment setup.
- Deploying on Hugging Face while supplying a real-time microphone input.

## 9.3 Next Steps (Future Enhancements)

- Improve noise handling in speech recognition.
- Fine-tune sentiment analysis model.

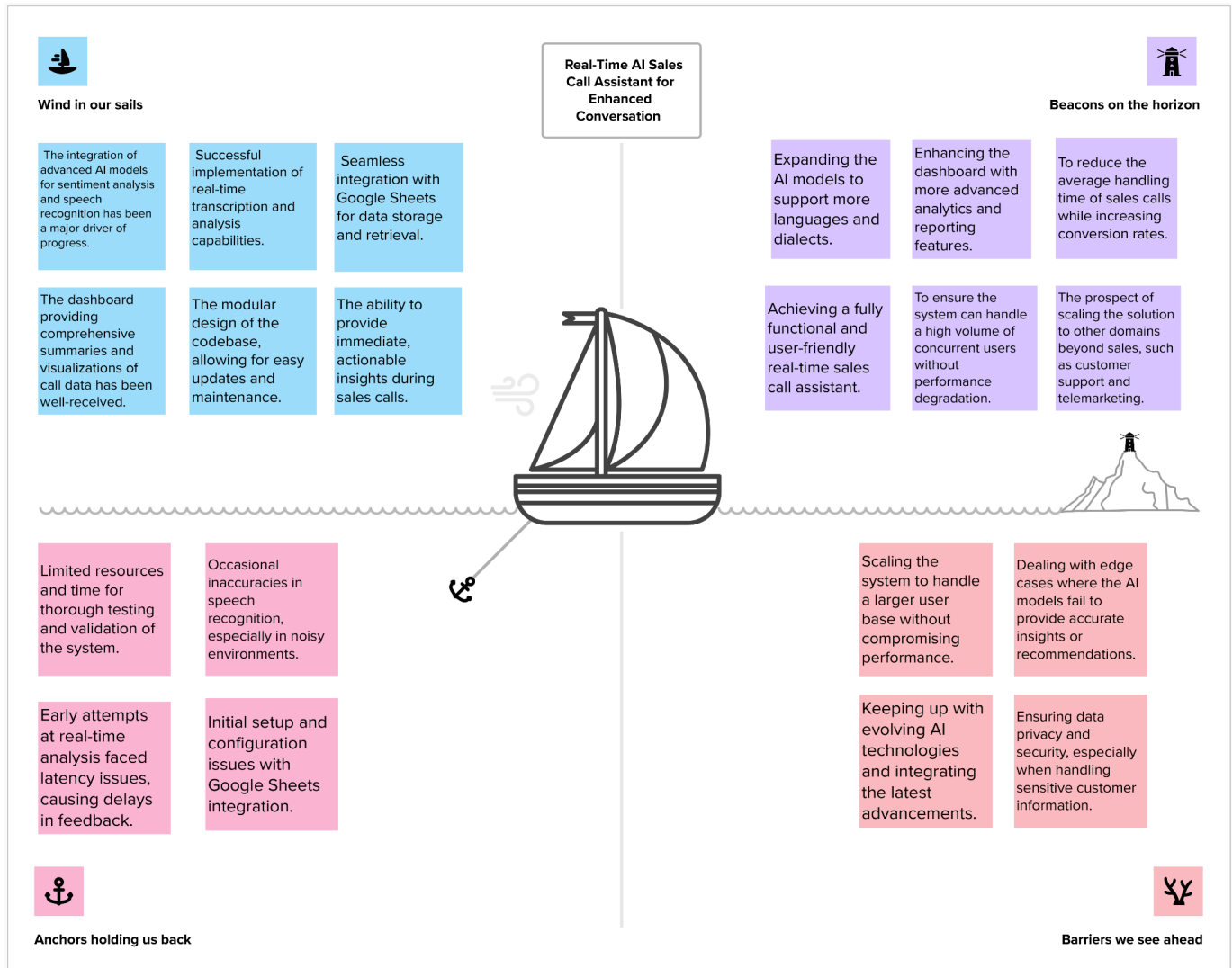


Figure 6. Sailboat Retrospective