Group Members

Quian Colao
Marc Louise Goco
Jessa Mae Lucido
Joshua Mercado

The software development life cycle model that we chose for our project is agile. We chose an agile model because our app has a modular design. Our app has many features that can be built as a module and each module can be tested iteratively. Furthermore, our system has many complex components and dividing it into smaller components is crucial. By developing each feature iteratively we can test each module running on our platform.

The data gathering and planning phase is crucial for the RoboAdvisor project because it helps identify the specific needs of investors who will use the app. During this phase, you collect detailed information about user preferences, investment goals, risk profiles, and trading behaviors, which are essential to personalize financial recommendations. This data forms the foundation for designing machine learning models and selecting the right financial indicators for analysis tools. Planning also involves defining technical requirements, such as API integrations with yfinance and Hugging Face, which are critical for real-time data access and AI recommendations. Thorough planning ensures that the project scope is clear, reducing the risk of scope creep and enabling effective sprint goal setting.

The design phase in RoboAdvisor focuses on creating intuitive and user-friendly interfaces for complex financial tools like stock analysis, portfolio management, and recommendation systems. The design process involves mapping out how users will interact with different components, such as entering stock symbols, viewing trend charts, and communicating with the NLP-powered chatbot. System architecture design is also part of this phase, determining how data flows between the backend, machine learning models, and the frontend display. Proper design ensures that technical integrations, like database management and API calls, are seamless and efficient. This phase balances both the functional and visual aspects to make the app accessible to users with varying levels of financial knowledge.

The development phase is where the RoboAdvisor platform truly takes shape by implementing the planned features and functionalities. This phase includes coding the backend services to fetch and store stock data, developing algorithms for trend and volatility analysis, and integrating machine learning models for price forecasting and recommendations. The development team also builds the user interface components, ensuring users can easily input data, view analysis, and manage their portfolios. During this phase, connecting the NLP chatbot to the portfolio management system allows users to interact naturally with the app. Continuous collaboration within the team helps resolve challenges quickly and incorporate new requirements as they arise.

Testing phase is essential to validate the accuracy, reliability, and usability of the RoboAdvisor's complex features. This involves running unit tests on individual modules such as data retrieval, analysis calculations, and ML predictions to ensure each works correctly. Integration testing confirms that components like the chatbot, database, and APIs function smoothly together without data loss or errors. User acceptance testing gathers feedback from real users, ensuring that the interface is intuitive and that recommendations meet investor needs. Thorough testing helps identify bugs early, improves the overall quality, and builds confidence before deployment.

Deployment phase brings the RoboAdvisor app to life by releasing it to a live or staging environment where users can access and interact with the platform. This phase involves setting up hosting, configuring servers, and ensuring secure access to sensitive financial data. Proper deployment allows real-time stock data updates and machine learning recommendations to operate without interruption. Monitoring tools are often implemented to track performance and detect any issues quickly after launch. Successful deployment marks the transition from development to user engagement, where the app starts delivering real value.

Review phase is the phase where you evaluate the overall success of the RoboAdvisor project and identify areas for improvement. Gathering feedback from users, stakeholders, and team members helps assess whether the app meets its intended goals, such as personalized investment advice and ease of use. Reviewing system performance data can reveal bottlenecks or accuracy gaps in machine learning models that need refinement. This phase also involves planning for future iterations or feature enhancements based on insights gathered. Continuous review fosters a culture of improvement, ensuring RoboAdvisor evolves to better serve its users over time.

Diagram

Phase	**Sprint 1**	**Sprint 2**	**Sprint 3**	**Sprint 4**	**Sprint 5**	**Sprint 6**
Phase 1: Data Gathering and Planning	Gather user requirements for stock analysis tools, portfolio management, and recommendations.	Identify necessary data fields and finalize the database design for stock data and portfolios.	Define required trend analysis indicators (moving averages, regression) and data sources.	Identify key volatility metrics (standard deviation, ATR) and determine how to calculate them.	Plan for the forecasting model (e.g., ARIMA) and gather necessary historical data	Finalize forecasting model selection and prepare historical stock data.
Phase 2: Design	Define the system architectur	Design database schema	Design the user interface	Design the user interface	Design the input interface	Design the visualization for

	e, database schema, and initial wireframes for the app.		for inputting stock symbols and time ranges, and displaying price trend charts.	for volatility input and chart visualization.	for stock symbol, time range, and forecast horizon.	forecasted vs. historical stock prices.
Phase 3: Development	Set up project repository, environment, and backend architecture for data storage.	Implement database tables, integrate the Yahoo Finance API for data retrieval, and ensure proper data storage.	Implement price trend analysis, including data retrieval, preprocessing, and trend indicator calculations.	Develop volatility calculation features, integrate with the database, and generate volatility charts.	Implement data retrieval and preprocessing for forecasting, and prepare models for training.	Train the forecasting model, generate predictions, and create visualizations for output.
Phase 4: Testing	Ensure that initial project setup and database connections work correctly.	Test the data retrieval process, ensuring data is pulled and stored correctly.	Test the trend calculation accuracy and visualizations with various stock symbols.	Test volatility calculations and ensure correct chart visualizations.	Test the preprocessing pipeline and verify data integrity for forecasting.	Test the accuracy of forecasts and visualize forecast confidence intervals.
Phase 5: Deployment	Deploy the basic project structure and backend to a local environment for testing.	Deploy the database structure and API integration to a staging environment.	Deploy the tool to the app’s staging environment for user testing.	Deploy volatility analysis tool to the staging environment.	Deploy the data retrieval and preprocessing components.	Deploy forecasting tool with interactive charts to the staging environment.
Phase 6: Review	Review requirements and	Review database setup and	Review the tool's accuracy,	Review the tool's functionalit	Review the model selection	Review forecast accuracy,

	project structure to confirm alignment with user needs.	data retrieval process	performance, and user interface for improvements.	y, ensure accurate volatility measurements	process and ensure data quality for forecasting.	UI, and model performance to refine predictions.

Phase	Sprint 7	Sprint 8	Sprint 9	Sprint 10	Sprint 11	Sprint 12
Phase 1: Data Gathering and Planning	Collect user requirements for portfolio management and how users will interact with it.	Define chatbot interactions for portfolio management and gather NLP model requirements.	Gather user preference data and identify stock data sources for the recommendation engine.	Plan and select appropriate machine learning models for personalized recommendations.	Gather feedback on user interface preferences for charts, dashboards, and recommendation details.	Collect bug reports and feedback from previous testing phases.
Phase 2: Design	Design the interface for portfolio creation, deletion, and stock management.	Design the conversational flow for portfolio management and NLP integration.	Design the process for integrating user data and stock data to generate personalized recommendations.	Design the recommendation model architecture, including ranking, scoring, and action suggestions.	Design UI components for displaying stock trends, volatility, forecasts, and recommendations.	Plan final refinements to features, UI, and database interactions based on testing feedback.
Phase 3: Development	Implement basic CRUD operations for portfolio management and database integration	Implement chatbot functionality for portfolio creation, updates, and deletions.	Implement data retrieval, cleaning, and preprocessing for the recommendation engine.	Implement the machine learning model, train it, and generate insights based on user portfolio	Implement interactive charts and dashboard components for better visualization	Implement final bug fixes, optimizations, and documentation for deployment.

				data.		
Phase 4: Testing	Test portfolio creation and update functions to ensure correct data manipulation.	Test chatbot interactions and ensure correct data manipulation in the database.	Test data integration and ensure that fresh data is available for recommendations.	Test recommendation accuracy, scoring, and user-specific insights.	Test UI responsiveness, usability, and ensure smooth navigation between tools.	Conduct comprehensive end-to-end testing, including performance, security, and usability tests.
Phase 5: Deployment	Deploy the portfolio management tool to a staging environment for internal testing.	Deploy the chatbot to the staging environment for real-world testing.	Deploy data integration components to a staging environment	Deploy the recommendation engine with live insights and recommendations to the staging environment.	Deploy visualizations and enhanced UI to the staging environment.	Deploy the complete application to the production environment.
Phase 6: Review	Review portfolio features and UI for user-friendliness and accuracy.	Review chatbot effectiveness, usability, and NLP accuracy.	Review the data flow and quality to ensure the recommendation engine will function smoothly.	Review the recommendation quality and refine the ML model based on test feedback.	Review the overall user experience and UI consistency for clarity and ease of use.	Review the entire project for any last-minute adjustments, finalize documentation, and prepare for closure.