**Quadratic Weather Modeling System Using the Agile Process Model**

**Author:** Rajarshi Ghosh  
**Email:** rghosh4@gitam.in  
**Course:** Software Engineering Lab — Assignment 2

**🌦️ Introduction**

The **Weather Modeling System** applies a **quadratic polynomial model** to predict three weather parameters at any given hour:

* Temperature (°C)
* Humidity (%)
* Rainfall intensity (mm/hr)

This implementation follows the **Agile Process Model**, which emphasizes iterative, incremental development, close collaboration with stakeholders, and adaptability to changing requirements. The system was developed in short sprints, delivering functional increments of the system at the end of each sprint.

**🔄 Sprint 1**

**📌 Sprint Planning**

* Define core functionality: predicting temperature using a quadratic model.
* Decide to use hardcoded data for initial implementation.
* Timeboxed sprint to deliver MVP (Minimum Viable Product).

**📌 Development**

* Design quadratic equation: .
* Implement Python script using numpy for curve fitting.
* CLI interface to display equation and predict temperature for a specific hour.

**📌 Review**

* Demo to users, collecting feedback on usability and results.

**📌 Retrospective**

* Identified improvements: add more parameters (humidity and rainfall) and user input mode.

**🔄 Sprint 2**

**📌 Sprint Planning**

* Incorporate humidity and rainfall predictions.
* Allow users to input data interactively.
* Add full-day forecast display.

**📌 Development**

* Extend input module for keyboard entry.
* Implement quadratic models for humidity and rainfall.
* Enhance CLI output with equations for all three parameters and full-day forecast.

**📌 Review**

* Users tested and provided feedback.

**📌 Retrospective**

* Users requested file upload option for ease of use and saving forecasts to a file.

**🔄 Sprint 3**

**📌 Sprint Planning**

* Implement CSV file upload feature (compatible with Colab/Jupyter).
* Add menu-driven interface for switching between input modes.
* Enable saving forecast output to .csv file.

**📌 Development**

* Enhance data handling to read and validate CSV input.
* Improve CLI with flexible navigation.
* Implement saving predictions to .csv.

**📌 Review**

* Delivered version with robust input options and flexible outputs.

**📌 Retrospective**

* Users requested graphical visualization, .xlsx support, and improved error handling.

**📊 Conclusion & Future Enhancements**

By applying the agile model, the Weather Modeling System evolved incrementally into a functional and adaptable application supporting multiple input modes, accurate predictions, and flexible output options while incorporating continuous user feedback.

**Future Improvements:**

* Graphical output (charts) for trends
* Support .xlsx files
* GUI-based version
* Enhanced validation and error messages