**Quadratic Weather Modeling System: Using the Iterative Process Model**

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

**Introduction**

**This document details the development of a Weather Modeling System that employs a quadratic polynomial model to predict three key weather parameters:**

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

**Our development process strictly followed the Iterative Process Model. This approach involved multiple cycles, each encompassing communication, planning, modeling, construction, and deployment. Every cycle refined the system based on crucial user feedback and rigorous testing, ensuring a continuously improving and robust application.**

**Iteration 1: Establishing the Core**

**The first iteration focused on building a foundational system for temperature prediction.**

**1.1 Communication**

* **Project Objective Defined: Predict temperature using a quadratic model.**
* **Initial Scope Set: Temperature prediction only, using hardcoded data for simplicity.**

**1.2 Planning**

* **Data Points Identified: Three predefined data points for model fitting.**
* **Interface Design: A basic Command-Line Interface (CLI) for single-hour predictions.**

**1.3 Modeling**

* **Quadratic Equation Design: The core quadratic equation *Y = AX² + BX + C* was selected as the mathematical model.**
* **Model Implementation: System designed to fit the quadratic model to the hardcoded data points.**

**1.4 Construction**

* **Initial Coding: A foundational Python version was developed, leveraging the numpy library for numerical operations.**
* **Output: The fitted quadratic equation and the temperature prediction for a specified hour were displayed.**

**1.5 Deployment**

* **The initial functional version was delivered to stakeholders for their feedback.**

**1.6 Feedback Received**

* **Expand Parameters: Request for support for humidity and rainfall predictions.**
* **User Input Required: Need to replace hardcoded data with user-provided input.**

**Iteration 2: Expanding Functionality and Input**

**Building on the first iteration, this cycle integrated additional weather parameters and enhanced user interaction.**

**2.1 Communication**

* **Feedback Integration: User feedback from Iteration 1 was thoroughly reviewed and prioritized.**
* **Scope Expansion: Confirmed the necessity of modeling humidity and rainfall.**

**2.2 Planning**

* **Input Module Enhancement: Designed to accept keyboard input from the user.**
* **Output Refinement: Output now includes all three predicted weather parameters.**

**2.3 Modeling**

* **New Models Developed: Dedicated quadratic models for humidity and rainfall were developed and integrated.**
* **UI Update: User interface redesigned to accommodate input for multiple parameters.**

**2.4 Construction**

* **Enhanced Codebase: The system was updated to support keyboard input mode.**
* **Multi-Parameter Prediction: Predictions for temperature, humidity, and rainfall were implemented.**
* **Full-Day Forecast: Added functionality to display a full-day weather forecast.**

**2.5 Deployment**

* **The updated system was released for user testing and further evaluation.**

**2.6 Feedback Received**

* **File Input: Request for functionality to upload data via files (e.g., CSV).**
* **Save Forecasts: Demand for the ability to save forecast results to a file.**

**Iteration (n): Robustness and User Experience**

**This ongoing iteration focuses on incorporating advanced input methods, flexible navigation, and persistent output options.**

**3.1 Communication**

* **Comprehensive Feedback Review: Discussed and prioritized feedback from Iteration 2.**
* **User Needs Confirmed: Strong user desire for file input and intuitive menu navigation.**

**3.2 Planning**

* **CSV Upload Support: Planned the integration of CSV file upload, particularly for environments like Google Colab or Jupyter notebooks.**
* **Menu-Driven Interface: Designed a robust menu system to allow users to easily switch between different input modes.**
* **Output Persistence: Planned for saving forecast output to a .csv file format.**

**3.3 Modeling**

* **Data Handling Enhancement: Developed robust data handling mechanisms for processing CSV input.**
* **Output File Format: Defined the precise structure and format for the output .csv file.**

**3.4 Construction**

* **Multi-Mode Implementation: Successfully implemented hardcoded, keyboard, and file upload input modes.**
* **Interactive Menu: Developed a comprehensive menu system, enabling users to switch modes and save their results.**

**3.5 Deployment**

* **The refined system, boasting significantly enhanced functionality and user experience, was delivered.**

**3.6 Feedback Anticipated / Future**

* **Graphical Representation: Request for graphical plots to visualize weather trends.**
* **Extended File Support: Support for .xlsx file input/output.**
* **Improved Error Handling: Enhanced validation and more informative error messages.**

**Conclusion & Future Enhancements**

**By diligently applying the Iterative Process Model, the Weather Modeling System has matured into a user-friendly and robust application. It now supports multiple input modes, provides accurate predictions for key weather parameters, and offers flexible output options.**

**Future Improvements will focus on:**

* **Graphical Output: Implementing charts and graphs for clearer visualization of weather trends.**
* **Expanded File Compatibility: Adding support for .xlsx file input and output.**
* **GUI Development: Transitioning to a Graphical User Interface (GUI) for a more intuitive user experience.**
* **Enhanced Validation: Implementing more sophisticated input validation and detailed error messages to improve system reliability.**