

PROJECT : Weather Prediction Model

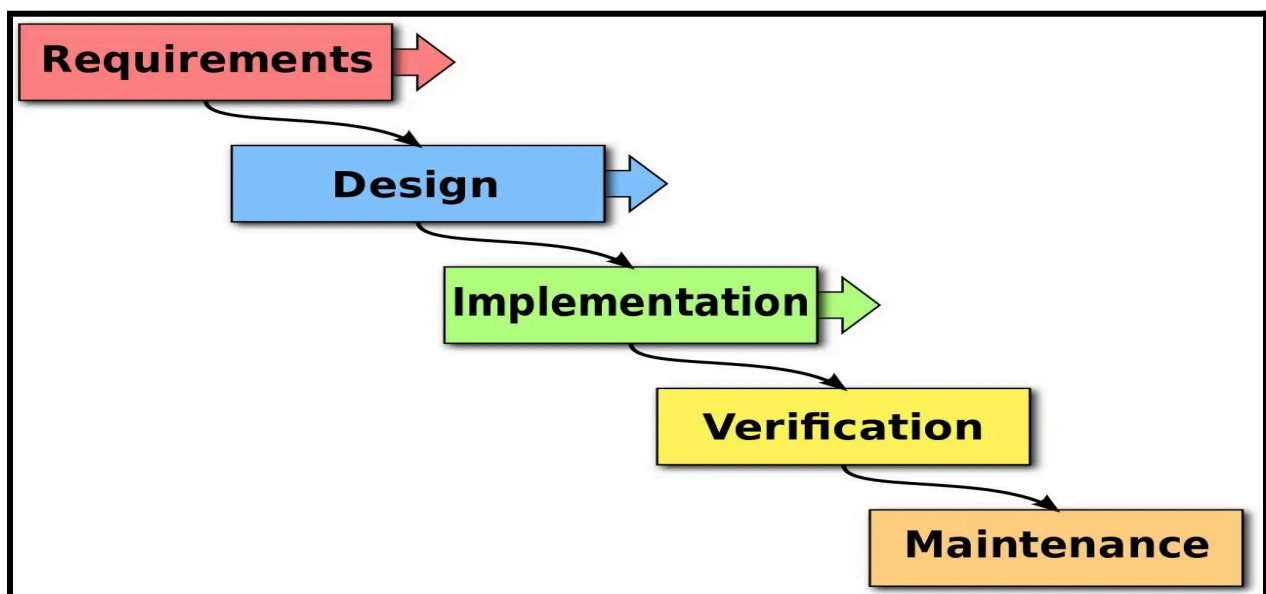
Introduction

Weather prediction systems rely heavily on accurate data analysis, computational models, and continuous feedback loops to forecast atmospheric conditions. This document provides detailed information about the development of a weather prediction model using the Waterfall methodology. The process is divided into five distinct phases: Requirements, Design, Implementation, Verification, and Maintenance.

Waterfall Methodology Overview

The Waterfall Method is a traditional software development process that consists of distinct, sequential phases. These phases are typically:

1. **Requirements**
2. **Design**
3. **Implementation**
4. **Verification**
5. **Maintenance**



Step 1: Requirements Gathering

- Hardware: RAM 8GB+, multi-core processor, servers, cloud (AWS/Azure).
- Software: OS (Windows/Linux), APIs, visualization tools.
- Data: Temp, wind speed, humidity, pressure from weather stations, IoT, satellites, radars.

Step 2: System Design

- **Architecture Design:**

High-level language, cloud vs on-premise setup.

- **Model Design:**

Model selection, distribution flow, calibration.

- **Data Processing Model:**

Data ingestion, preprocessing, integration.

Step 3: Implementation

Weather Model Integration:

- Implement weather modeling.

Data Handling:

- Collection, validation, ingestion.

Backend Development:

- Backend infrastructure setup.

Frontend Development:

- Front-end integration.

Testing and Debugging:

- Unit testing, integration testing, performance testing.

Step 4: Validation and Verification

Model Testing:

- Accuracy testing.

System Testing:

- End-to-end validation.

Step 5: Deployment

Infrastructure Setup:

- Server configuration.

Deployment Strategy:

- Monitoring and logging.

Documentation:

- User and technical manuals.

Step 6: Maintenance and Updates

Model Improvement:

- Retraining and updates.

System Maintenance:

- Bug fixes and optimization.

Step 7: Post-Deployment Monitoring and Feedback

Performance Monitoring:

- System health tracking.

User Feedback:

- Continuous improvements.

