

Software Engineering Task-2

Kavya Aggarwal

HU22CSEN0100288

1. Develop weather modeling using the quadratic model using waterfall technique.
2. Write about all the phases of the waterfall model

1. Importing Necessary Libraries

```
✓  
0s [18] import numpy as np  
      import matplotlib.pyplot as plt
```

2. Defining the Quadratic Expression

```
✓  
0s [19] def quadratic(x, a=-1, b=0, c=10):  
      return a * x ** 2 + b * x + c
```

3. Defining The stages of Waterfall model and implementing it using pyplot library.

```
stages=["Requirements","Design","Implementation","Testing","Deployment","Maintenance"]
x_positions = np.linspace(-2, 2, len(stages))
y_positions = quadratic(x_positions)

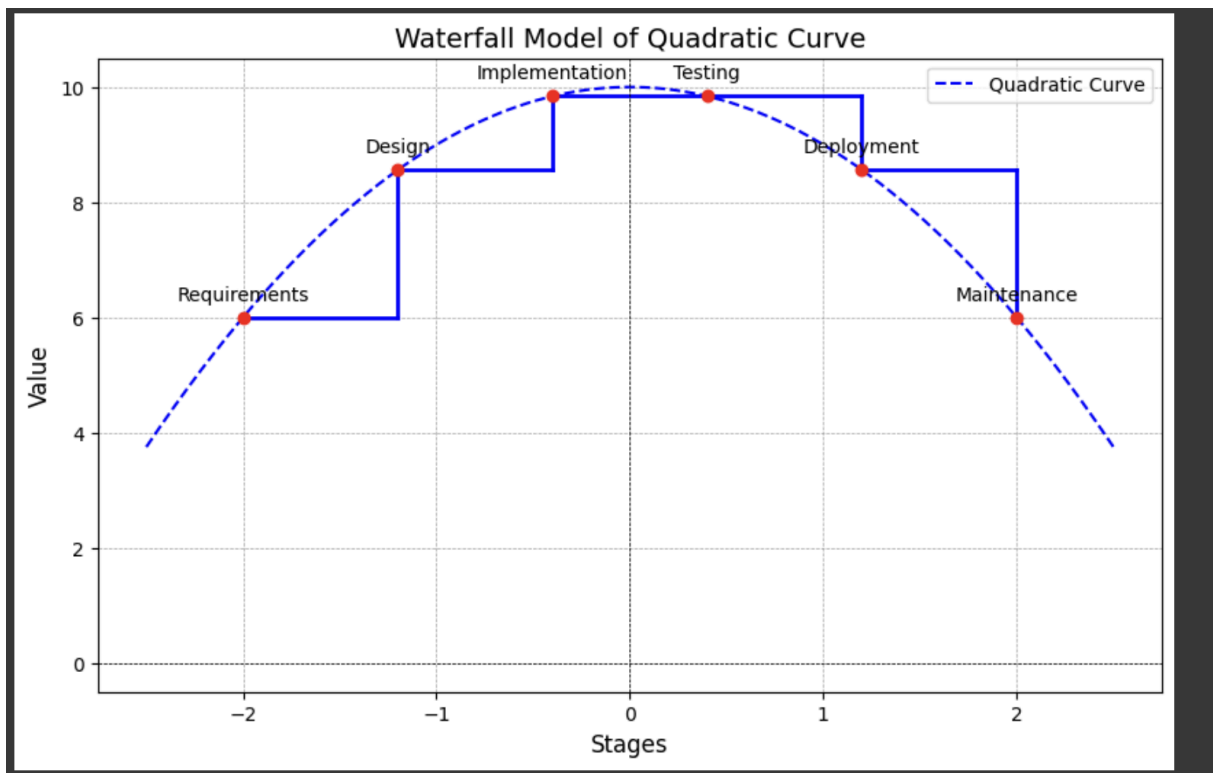
#Plotting the Quadratic Curve
x_curve = np.linspace(-2.5, 2.5, 500)
y_curve = quadratic(x_curve)
plt.figure(figsize=(10, 6))
plt.plot(x_curve, y_curve, color="blue", linestyle="—", label='Quadratic Curve')

# Adding the markers and connecting them with the vertical lines to create the waterfall effect
for i in range(len(stages)):
    if i>0:
        plt.plot([x_positions[i-1], x_positions[i]], [y_positions[i-1], y_positions[i]], color="blue", linewidth=2) #Horizontal Segment
        plt.plot([x_positions[i], x_positions[i]], [y_positions[i-1], y_positions[i]], color="blue", linewidth=2) #Vertical Drop

#Placing the markers and labels
plt.scatter(x_positions[i], y_positions[i], color="red", zorder=5)
plt.text(x_positions[i], y_positions[i] + 0.3, stages[i], ha="center", fontsize=10)

#Styling the plot
plt.title("Waterfall Model of Quadratic Curve", fontsize=14)
plt.xlabel("Stages", fontsize=12)
plt.ylabel("Value", fontsize=12)
plt.axhline(0, color="black", linewidth=0.5, linestyle="—")
plt.axvline(0, color="black", linewidth=0.5, linestyle="—")
plt.grid(color="gray", linestyle="—", linewidth=0.5, alpha=0.7)
plt.legend()
plt.show()
```

4. Output of the Implementation



Phases of the Waterfall Model

The Waterfall Model is a sequential design approach often used in software development processes. It is characterized by its structured progression through distinct phases, where the completion of one phase marks the beginning of the next. Below are the key phases of the Waterfall Model:

1. Requirement Analysis

- **Objective:** Gather and document all requirements of the system or software from stakeholders.
- **Activities:**
 - Conduct interviews, surveys, and meetings with stakeholders.
 - Analyze user needs and document them in a Software Requirement Specification (SRS) document.
- **Outcome:** A clear and comprehensive understanding of the project's scope and objectives.

2. System Design

- **Objective:** Translate the requirements into a design blueprint for the system.
- **Activities:**
 - Define the system architecture and interfaces.
 - Create design documents, including data flow diagrams, system models, and entity-relationship diagrams.
 - Specify hardware, software, and network requirements.
- **Outcome:** A detailed system design document that guides developers during implementation.

3. Implementation

- **Objective:** Convert the system design into actual code.
- **Activities:**
 - Write and compile code modules as per the design.
 - Integrate different modules to build the complete system.
 - Perform unit testing to verify each module.
- **Outcome:** A fully functional software product ready for integration and testing.

4. Integration and Testing

- **Objective:** Ensure that all components work together as intended and the software meets the requirements.
- **Activities:**
 - Integrate all modules and components.
 - Conduct various tests (e.g., functional, performance, security, and usability testing).
 - Identify and fix defects or inconsistencies.
- **Outcome:** A thoroughly tested system with minimized defects and confirmed functionality.

5. Deployment

- **Objective:** Deliver the completed product to the end users.
- **Activities:**
 - Install and configure the software in the production environment.
 - Provide training and documentation for users.
 - Monitor initial use and address any issues that arise.
- **Outcome:** The software is operational and available for users.

6. Maintenance

- **Objective:** Ensure the software remains functional and relevant post-deployment.
- **Activities:**
 - Fix bugs or issues reported by users.
 - Implement updates and enhancements as needed.
 - Monitor system performance and ensure compatibility with evolving technologies.
- **Outcome:** Sustained system performance and user satisfaction over time.

Characteristics of the Waterfall Model

- **Linear Sequential Flow:** Each phase must be completed before the next begins.
- **Documentation-Driven:** Extensive documentation is produced at every phase.
- **Rigid Structure:** Minimal flexibility to accommodate changes once a phase is completed.





Advantages

- Simple and easy to understand and manage.
- Clearly defined stages and deliverables.
- Ideal for smaller projects with well-understood requirements.

Disadvantages

- Inflexible to changing requirements.
- Late detection of issues, as testing occurs after implementation.
- Not suitable for complex or long-term projects with evolving needs.

5. Pushed to GitHub

| | | | |
|--|-----------------------------|-------------------------|---------------|
|  OmegaKavya | Software_Engineering_task_2 | 0d00f7d · 2 minutes ago | 🕒 3 Commits |
|  SE_Exp_1.ipynb | Software_Engineering_task_1 | | 2 weeks ago |
|  SE_Exp_2.ipynb | Software_Engineering_task_2 | | 2 minutes ago |
|  weather_modeling.txt | Add files via upload | | 2 weeks ago |

