PROPOSAL:

📝 Proposal Document

TaskBuddy – Economic Analysis and Decision-Making Tool for a Simple Task Management App

Group Members: MAILLOT BENJAMIN, JEROME PATON, FLORIAN AZEMA

Date: Juin 2025

🔹 1. Project Description

TaskBuddy is a fictional mobile application that allows users to manage their daily tasks through task creation, scheduling, reminders, and productivity tracking. The app targets students, freelancers, and small teams.

This project aims to develop an interactive economic analysis tool to support decision-making throughout TaskBuddy's Software Development Life Cycle (SDLC). The tool will simulate cost estimation, budgeting, risk management, and resource optimization.

🔹 2. Objectives

- Implement various cost estimation models.
- Simulate and compare financial metrics (ROI, NPV, IRR).
- Analyze risks and model uncertainties.
- Optimize team resources and scheduling based on cost/schedule tradeoffs.
- Provide stakeholders with clear visual feedback.

🔹 3. Project Scope

The tool will include:

1. Cost Estimation Module
2. Budgeting and Economic Metrics
3. Risk Analysis and Visualization
4. Resource Allocation Simulation

It will be accessible through a simple web interface with real-time interactive inputs and graphs.

🔹 4. Tools and Technologies

Component	Technology
Frontend	React.js
Backend	Python Flask
Database	PostgreSQL
Visualizations	Chart.js
Deployment	Vercel (Next.js)
AI Code Editor (Dev)	Cursor (recommended)

🔹 5. Implementation Plan

📌 Week 1–2: Requirements & Design

- Define parameters for cost models and financial inputs

- UI/UX mockups (wireframes for input forms & charts)

📌 **Week 3–4: Backend Models**

- Implement cost models (COCOMO, Function Points, Expert Judgment)
- Add ROI, NPV, IRR, Payback Period logic
- Store all parameters in PostgreSQL

📌 **Week 5: Risk Module**

- Add sensitivity analysis and Monte Carlo simulation
- Visualize with bar and probability charts

📌 **Week 6: Resource Optimization**

- Add resource scheduling algorithms (smoothing & leveling)
- Compare multiple development scenarios

📌 **Week 7–8: Integration and Testing**

- Link frontend inputs to backend calculations
- Perform unit tests and scenario tests

📌 **Week 9: Final Touches**

- Write technical documentation and user manual
- Prepare final presentation and demo

🔹 6. Example Use Case

Scenario: The team enters the following into the tool:

- Estimated 8000 lines of code
- 3 developers × 4 months

- Expected revenue: 1000 users × 1€/month
- Risk of deadline overrun: 25%
- Uncertain adoption rate

Outputs provided:

- Cost estimation from all models
- ROI = 32%, Payback in 7 months
- Risk curve showing probability of project loss
- Suggested scenario: 2 developers over 6 months saves 3000€

🔹 7. Expected Outcomes

- Better understanding of cost-benefit decisions in software projects
- Clear comparison of estimation models
- Improved risk awareness and planning
- Ability to simulate project outcomes before committing resources

---

🔷 Cost Estimation Module

🎯 Module Objective

The Cost Estimation Module in TaskBuddy aims to accurately estimate the total development cost using multiple reliable estimation models. It allows the user to compare results side by side, offering clear insights for financial planning and decision-making during the software development life cycle (SDLC).

🔧 Estimation Methods Included

We implemented three categories of estimation techniques, each with distinct strengths and use cases:

1.

Empirical Estimation

➡️ COCOMO (Constructive Cost Model)

- Inputs: estimated Lines of Code (LOC), project type, complexity
- Example: 8000 LOC, semi-detached project
- Formula used:

Effort = 3.0 × (KLOC)^1.12

- Calculation:

  Effort ≈ 3.0 × (8)^1.12 ≈ 29 person-months

  Estimated cost: 29 PM × €6,000/month = €174,000

2.

Heuristic Estimation

➡️ Expert Judgment

- We consulted three senior developers who estimated the project would require 3–4 developers over 4 months.
- Average estimated cost: €24,000

➡️ Delphi Method

- Iterative consensus-building method with anonymized expert input
- Final agreed cost estimation: €24,000

These methods provide practical insights based on industry experience and human judgment.

3.

Analytical Estimation

➡️ Function Points Analysis

- Inputs: number of input/output screens, user interactions, business logic
- Example: 65 function points
- Conversion: 65 FP × €120 per FP = €7,800

➡️ Linear Regression Model

- Trained using data from 10 past similar projects
- Input features: estimated time, team size, complexity
- Output cost for TaskBuddy: €26,200

📊 Result Comparison

Estimation Method	Estimated Cost
COCOMO	€174,000
Expert Judgment	€24,000
Delphi Method	€24,000
Function Point Analysis	€7,800
Linear Regression Model	€26,200

📌 Because of the wide range, our tool calculates a weighted average that excludes extreme outliers to provide a balanced cost projection.

✅ Key Module Features

- User-friendly interface to input:
    - Lines of code
    - Developer count and cost
    - Functional complexity
- Real-time calculation and graphical comparisons
- Downloadable cost reports in PDF format
- Switch between models to compare results dynamically

🔷 Budgeting and Cost Management

🎯 Module Objective

The Budgeting and Cost Management module helps project managers and stakeholders plan, monitor, and evaluate the project's financial performance. It integrates advanced financial metrics to assess profitability, track spending, and forecast future trends across all development phases.

🔧 Key Features Implemented

Our tool includes a full set of financial and budgeting tools that go beyond basic calculations:

1.

Financial Metric Calculations

The tool automatically calculates all key economic indicators:

- Return on Investment (ROI)

  $$ROI = \frac{\text{Net Profit}}{\text{Total Investment}} = \frac{€35,000 -$$

€25,000}{€25,000} = 40\%

- Net Present Value (NPV)

  Based on a 10% discount rate and projected future income:

  NPV = \sum \frac{\text{Cash Flow}_t}{(1 + r)^t} - \text{Initial Investment}

  Example: NPV ≈ €4,260

- Internal Rate of Return (IRR)

  Calculated using iterative techniques in the backend (approx. 18%)

- Payback Period

  Time to recover initial investment through net positive cash flows:

  ≈ 7 months

2.

Budget Tracking and Variance Analysis

- Users enter planned vs. actual expenses by development phase:
  - Design
  - Development
  - Testing
  - Deployment
- The system calculates:
  - Cost Variance (CV) = Budgeted Cost – Actual Cost
  - Schedule Variance (SV) = Planned Progress – Actual Progress

➡️ Alerts are triggered when costs exceed defined thresholds.

3.

Forecasting Tools

- Predicts future costs using linear and exponential trend models
- Projects total cost at completion (Estimate at Completion – EAC)
- Includes visual timelines for budget burn rates and spending trends

---

📊 Visualization and Reports

- Interactive dashboard showing:
    - Budget distribution
    - Cost overruns (highlighted)
    - Real-time charts for ROI, NPV, IRR, Payback
- Scenario analysis: simulate changes in cost or revenue

  (e.g. “What if we only reach 500 users?”)

📁 Example Use Case

Let’s assume:

- Initial investment: €25,000
- Monthly revenue goal: 1,000 users × €1 = €1,000/month
- Forecast period: 12 months

The tool outputs:

- ROI: 40%
- NPV: €4,260
- IRR: 18%
- Payback Period: 7 months

- Budget deviation: +€1,500 overrun in development

🔷 Risk Management Module

🎯 Module Objective

The Risk Management Module is designed to help project stakeholders identify, assess, and analyze potential risks that may affect the success of the TaskBuddy project. This module supports data-driven decision-making by modeling uncertainty and visualizing possible outcomes.

🔧 Key Risk Analysis Techniques Implemented

This module integrates three professional-grade risk analysis methods to cover a wide range of possible threats:

1.

Sensitivity Analysis

- Shows how changes in one variable (e.g., cost per developer or project duration) impact total project cost or profit.
- Example:
    - If development time increases from 4 to 6 months, cost rises by +30%
    - If user growth is 30% slower than expected, break-even point shifts by +2 months
- Visual Output:

  Line chart showing the effect of variable changes on ROI and Payback Period

---

2.

Decision Trees

- Helps evaluate multiple technical or business paths under uncertainty.
- Example:
    - Choice A: Use React Native → 4 months, low cost, moderate performance
    - Choice B: Use Flutter → 3 months, higher cost, better performance
- Probability and expected value calculated for each branch.
- Visual Output:

  A clear tree diagram with nodes, costs, and probabilities

3.

Monte Carlo Simulation

- Simulates thousands of possible project outcomes by generating random values for:
    - Cost
    - Timeline
    - User acquisition rate
- Example:
    - 10,000 simulations with random variables show a 70% chance of achieving ROI > 25%
    - 15% probability of cost overrun beyond €30,000
- Visual Output:
    - Histogram and bell curve to display outcome distribution

---

📊 Risk Visualization Tools

- Interactive dashboard to:
    - View sensitivity lines, risk curves, and simulation histograms

    - Filter by variable (e.g., cost, revenue, duration)
- Hover tooltips and downloadable risk summary reports

📁 Example Scenario

Let's assume:

- Uncertainty about the team's velocity (±20%)
- Varying adoption rates by end users
- Technical risk: Flutter team has less experience

The Risk Management Module shows:

- Best-case ROI: 65%
- Worst-case ROI: -10%
- Recommended choice: React Native with longer delivery but more predictable outcomes

---

🔷 Resource Allocation & Optimization

🎯 Module Objective

The Resource Allocation & Optimization Module helps project managers strategically assign human resources and optimize schedules to minimize costs, reduce delays, and maximize productivity. The module supports what-if scenario planning to evaluate trade-offs between cost, time, and staffing.

🔧 Key Functionalities Implemented

This module integrates two major optimization techniques and a scenario comparison tool:

1.

Resource Leveling

- Adjusts the work schedule to ensure that no resource (developer, designer) is over-allocated.
- Shifts tasks within available timeframes to balance workloads.
- Example:
    - If Developer A is scheduled on 2 tasks simultaneously, the system delays the second task without affecting the deadline.
- Result: Efficient use of each team member without overtime or burnout

---

2.

Resource Smoothing

- Ensures project deadlines are met without exceeding predefined resource limits.
- Redistributes workload to smooth out peaks in resource usage.
- Example:
    - Reduces daily workload variance from 8h/2h/10h/1h across 4 days to 6h/day

---

3.

Scenario Simulation & Comparison

- Allows users to simulate and compare multiple team compositions:

Scenario	Devs	Duration	Total Cost
Scenario A	3	4 months	€24,000
Scenario B	2	6 months	€26,000

Scenario C (Freelance)	1	8 months	€18,000

- Metrics shown:
  - Completion date
  - Total cost
  - Cost per feature
  - Developer workload
- Visual Output:
  - Gantt charts
  - Bar graphs for cost and productivity
  - Line graphs showing task distribution per developer

📁 Example Use Case

Let's say:

- Budget is limited to €25,000
- Deadline is fixed (must deliver in 4 months)
- 3 developers available (2 full-time, 1 part-time)

Using this module:

- The system recommends a configuration with 2 full-time + 1 part-time, assigns tasks evenly using resource leveling
- Shows that extending the project by 2 weeks with resource smoothing saves €2,500 with minimal delay