Comprehensive Documentation

林朝源 Melka，丁奕辰 Adrien Lopez，张建兴，王笑伟

1. Background

In the dynamic landscape of software engineering, effective economic decision-making throughout the Software Development Life Cycle (SDLC) is critical for project success. Software projects often face challenges such as inaccurate cost estimations, inadequate budget tracking, insufficient risk assessment, and suboptimal resource allocation, which can lead to cost overruns, schedule delays, and failed deliverables. Existing tools typically focus on isolated aspects of economic analysis (e.g., cost estimation or risk management) but lack integration of comprehensive models and interactive capabilities to support holistic decision-making.

The growing complexity of software projects, coupled with the need for data-driven insights, highlights the demand for a unified tool that integrates multiple economic models and analysis techniques. This gap in current solutions motivated the development of an interactive software tool designed to address these challenges. By combining empirical cost estimation methods, financial metric computations, risk assessment frameworks, and resource optimization algorithms, the tool aims to empower software engineering teams to make informed, data-backed decisions that balance economic viability with project objectives.

2. Project Overview

The "Economic Analysis and Decision-Making Tool for Software Projects" is an interactive web-based application developed to support software engineering teams in evaluating economic decisions throughout the SDLC. The tool serves as a comprehensive platform that integrates various economic models and analytical techniques, enabling stakeholders to address critical aspects of project management, including cost estimation, budgeting, risk management, and resource allocation.

3. Core Objectives

- Apply economic concepts (e.g., cost-benefit analysis, financial metrics, risk modeling) to practical software engineering scenarios.
- Provide hands-on experience in economic modeling, cost estimation, and data-driven decision-making.
- Develop analytical skills through the implementation and evaluation of integrated economic decision-making frameworks.

4. Deliverables

The project delivers a fully functional web application, technical documentation detailing model implementations, a user manual, and a demonstration of key scenarios (e.g., cost estimation comparisons, risk simulations, resource optimization). The tool enables users to input project parameters, generate comprehensive economic analyses, and visualize outcomes to support strategic decision-making throughout the software development process.

5. Key Functional Components

1. Cost Estimation Module: Implements multiple estimation techniques (COCOMO II, Function Point Analysis, Expert Judgment, Regression Analysis) to enable comparative cost and effort forecasting.
2. Budgeting and Cost Management: Offers tools for tracking budgets, analyzing variances, and computing financial metrics (ROI, NPV, IRR, Payback Period).
3. Risk Management Module: Integrates sensitivity analysis, decision trees, and Monte Carlo simulations to model uncertainty and visualize risk outcomes.
4. Resource Allocation & Optimization: Utilizes Critical Path Method (CPM), resource leveling, and scenario analysis to optimize resource scheduling and balance costs against schedules.

6. Technical Details

- Frontend: React with Tailwind CSS for an intuitive user interface and data visualization.
- Backend: Flask-based RESTful API for model computations and data processing.
- Database: MySQL for storing project metadata, user inputs, and historical cost data.
- Visualization: Chart.js and D3.js for interactive data representation.

6.1 Front End Implementation

6.1.1 DataCollection Component

The data collection interface is divided into four main sections:

1. Project Overview

- Project name input
- Basic project information
-

2. COCOMO Parameters

- SLOC (Source Lines of Code) input
- Project class selection (Organic, Semi-detached, Embedded)
- Effort Adjustment Factor (EAF) configuration

- Project Class Descriptions
    - Organic (O): Small teams, familiar environment, flexible requirements
    - Semi-detached (S): Medium teams, mixed experience, medium complexity
    - Embedded (E): Large teams, complex systems, tight constraints

3. Financial Settings

- Discount rate configuration
- Financial analysis parameters
- Used for NPV and financial calculations

4. Resource Planning

- Task management (add/remove tasks)
- Resource capacity configuration
- Dependencies between tasks

6.1.2 ResultsDashboard Component

The dashboard provides a comprehensive view of the analysis results through multiple tabs:

1. Overview Tab

- Key metrics display
- Cash flow analysis chart
- Resource utilization visualization
- Project summary

2. COCOMO Analysis Tab

- Effort estimation
- Duration calculation
- Team size requirements
- Cost estimation

3. Financial Analysis Tab

- ROI calculation
- NPV analysis
- IRR metrics
- Payback period

4. Risk Analysis Tab

- Risk score calculation
- Risk factors assessment
- Mitigation strategies

5. Resources Tab

- Resource allocation
- Team composition
- Cost distribution

6. History Tab

- Historical data visualization
- Trend analysis
- Comparative metrics

6.2 Backend Implementation

6.2.1 Implementation Description

The backend of this Flask project consists of several Python modules, each responsible for different functionalities:

1. budget.py:

- predict_project_viability: Combines historical data and new project parameters to predict project viability. It first calculates the development cost using COCOMO output and then generates cash flow predictions based on historical data. Finally, it calls calculate_financial_metrics to calculate financial metrics.
- calculate_financial_metrics: Calculates financial metrics such as ROI, NPV, IRR, and payback period based on cash flows and discount rate. It also generates cash flow analysis data and budget tracking data.
- get_history_budget: Generates historical data with a time dimension.

2. cocomo.py:

- cocomo: Performs COCOMO II analysis based on source lines of code (SLOC), project class, and environment adjustment factor. It calculates effort, duration, people, and cost in USD.

3. resource.py:

- get_mock_task_data: Returns mock task data and resource capacity.
- calculate_resource_allocation: Performs simplified resource allocation calculations, including critical path calculation, resource utilization simulation, and optimization simulation.

4. risk.py:

- calculate_risk_analysis: Calculates risk analysis results, including risk matrix and total risk score, based on COCOMO output.
- monte_carlo_simulation: Performs Monte Carlo simulation based on cash flow analysis data.
- perform_sensitivity_analysis: Performs sensitivity analysis on financial predictions.

In the Flask application, these functions can be integrated into API endpoints to handle incoming requests and return corresponding results. For example, the /history endpoint can call get_history_budget, and the /analyze endpoint can call functions from different modules to perform a full analysis.

6.2.2 Interface Documentation

See “API Interface Documentation.pdf” for more detailed information.