

**CUMULATIVE DELAY DURATION EQUATION (CDDE) FOR
MEASURING SOFTWARE PROJECT DELAYS AND COST
IMPACT**

**Eng. Ahmad Abdullah Alghamdi*, Prof. Abdullah Al-Qarni,
Dr. Wajdi Al Jedaibib**

Faculty of Computing and Information Technology, King Abdulaziz
University, P.O. Box 80200, Jeddah 21589, Saudi Arabia
Email: analghamdi@stu.kau.edu.sa, amsalgarni@kau.edu.sa,
waljedaibi@kau.edu.sa

Abstract

This study aimed to refine and validate the Cumulative Delay Duration Equation (CDDE) as a systematic model for tracking the delay in a software project and its costs. The sample included four projects: an ERP implementation project as part of an overall implementation plan, a surveillance camera system, a web portal project, and an infant protection system project. A mixed-method (survey questionnaires, expert interview and historical project documents review) method was implemented. Data was collected using project logs, stakeholder feedback and structured expert assessments. Quantitative analysis was conducted using descriptive statistics, regression modelling and comparison of delay duration methodology with models such as COCOMO and EVM to demonstrate how the CDDE diagnoses project delays. The study indicated that while the CDDE does not directly reflect the delay percentage, the CDDE does consistently consider the cumulative weighted impacts of management, resources and external elements to provide a more realistic basis to address the potential cost overruns. The study concluded that the CDDE needs to be validated across a broader spectrum of projects, integrated with actual project tools, and improved weighting aspects through a data-informed methodology, such as machine learning. The study suggested that CDDE could be a decision support tool to assist project managers in identifying, categorising, and managing delays that could improve prediction accuracy and eventually delay mitigation and cost containment.

Keywords: Software Project Delays, Cost Estimation, Cumulative Delay Duration Equation (CDDE).

1. Introduction

Software project delays, in today's project management practices, represent one of the most significant issues as project delays impact cost, schedule performance, and project success. Organizations in every sector (healthcare, government, education, logistics, etc) have become more dependent on advancing their digital transformation strategies, which include software implementations with increasing complexity. Despite advances in project management methodologies (waterfall project management, agile project management, hybrid project management methodologies, and data-driven project management—all of which have their

place), projects still experience unplanned delays, which can negatively affect budgets and decrease stakeholder satisfaction (Mahdi et al., 2021).

Project delays have impacts that extend beyond schedule delays and are often visible as cost overruns, inappropriate resource allocation, and possibly, reputational damage and external societal impacts. Traditional cost estimating models, such as the Constructive Cost Model (COCOMO), and estimating methods that rely on expert judgement, do not adequately accommodate the cumulative and compounding effects of project delay, where disruptions in the early stages lead to greater bottlenecks later (Ahmed et al., 2021). To help fill this gap, the Cumulative Delay Duration Equation (CDDE) was developed to systematically quantify and evaluate the effects of delay on project costs.

The CDDE is distinctive in that it gives not only a record of independent delay events but also the cumulative effect of these events across a project. For instance, if an incorrect estimate of the requirement stage has delayed the design and testing stages because decisions in design were necessarily reliant on decisions in the requirement definition, the cost-based implications can become exponential when new designs cause new delays, and delays that are not properly estimable by static estimating frameworks. CDDE shows a dynamic demonstration of schedule slippage and accompanying cost overruns that facilitate accurate forecasting and timely intervention (Kula et al., 2023).

The CDDE study has situated itself among wider theoretical viewpoints; Systems Theory, Contingency Theory, and the management perspective of Goal-Oriented Measurement (GQM), which address the interrelated, contextual and measured dimensions of measuring software projects. These theories support the premise that projects function as a dynamic and interrelated system, and that the delayed reaction to one area of a project impacts the whole project lifecycle, practically requiring delayed, contextual, and goal-oriented measurement to stretch out its processes (Mamedova & Bolonin, 2025). The application of CDDE in this context provides project leaders with a strong vehicle for delay-focused cost estimation, which will benefit all situations where time to completion and available resources are restricted.

2. Problem Statement

Delays in software projects are a universal and perennial issue, causing both serious cost overruns and poor quality outcomes. Despite years of advancements in methodologies, including agile processes, hybrid estimation approaches, and machine learning based predictions (Taboada et al, 2023), most projects, regardless of the approach, still do not stay within their intended schedule and budget. Research has shown that deviations in schedules impact not only the processes for specific projects, but they also lead to systematic alterations such as changing resources assigned to the project, team exhaustion, and expedited defect rates discovered during later stages of the project (Kuutila et al., 2020).

Traditional estimates always underestimate the compounding effect of delays. Poor planning or scope creep, for example, can snowball into a significant bottleneck with testing and introduce exponentially greater costs (Binu, 2025). Although models like the Total Delay Risk Index (TDRI) have been proposed in similar fields like construction (Koulinas et al., 2020),

very few tools explicitly measure cumulative delays in software projects. Without structured approaches, project teams respond to the changes, rather than prevent budget overruns.

The Cumulative Delay Duration Equation (CDDE) offers a resolution to the issue by isolating the cumulative effects of schedule changes across all project phases. However, empirical evidence on the CDDE in diversified sectors is limited. Few contexts offer challenges as critical as health systems, government digital transformation, education technology, and logistics management; software delays can equate to monetary pressures, and potentially carry societal impacts (Sepasgozar et al., 2019). Hence, the research problem is the lack of integrated, theoretically informed, and empirically validated approaches for measuring cumulative delay and its resultant costs. This paper articulates the CDDE as a systematic approach to bridging this gap.

Study Objectives

1. To create a systematic framework for applying the Cumulative Delay Duration Equation (CDDE) in the measurement of software project delays and estimation of costs.
2. To investigate how CDDE can be positioned as part of Systems Theory, Contingency Theory, and Goal-Oriented Measurement (GQM) in regards to the meaning of software project delays.
3. To empirically establish evidence for the use of the CDDE in health, government, education, and logistics sectors.
4. To propose sector-specific recommendations for mitigating cumulative project delays and minimizing financial risks.

Study Questions

1. How can CDDE be applied to measure cumulative delays and their cost implications in software projects?
2. In what ways does CDDE complement existing theoretical frameworks such as Systems Theory, Contingency Theory, and GQM?
3. What are the sectoral variations in software project delays and cost impacts when measured through CDDE?
4. How can CDDE-informed insights guide project managers in designing proactive delay mitigation strategies?

Study Hypotheses

H1: The application of CDDE significantly improves the accuracy of software cost estimation compared to traditional estimation models.

H2: Cumulative project delays, when quantified through CDDE, have a direct and positive correlation with cost overruns.

H3: Integration of CDDE with Systems Theory and Contingency Theory enhances predictive capacity for project risk management.

H4: The use of CDDE in sector-specific projects (health, government, education, logistics) yields differentiated but consistently valuable cost-control outcomes.

3. Literature Review and Theoretical Framework

Evolution of Software Project Management

The trajectory of software project management has evolved from linear, plan-driven models such as Waterfall to iterative methodologies such as Agile and Scrum, and more recently toward hybrid, data-driven, and machine-learning-enhanced frameworks (Mahdi et al., 2021). Early models emphasized sequential development, which struggled to adapt to changing requirements. Agile methodologies, in contrast, introduced iterative cycles, stakeholder collaboration, and continuous feedback to reduce uncertainty. But whilst agile reduced rigidity, it did not overcome the fundamental problems of delayed delivery and going over budget, particularly in large-scale projects and those with a distributed model (De Sousa et al., 2022). The use of probabilistic modelling and artificial intelligence (AI) as an example reflects the ongoing shift to predictive and adaptive estimation. Companies have started to use Bayesian Belief Networks (BBN) and neural networks to predict the risk of delay and cost, by including prior knowledge and historical data into estimation (Mahmood et al., 2022). While helpful, a significant number of delays continue (meaning change is needed), and delays are often significant. We need frameworks that consider the cumulative impact of delays, not just the impact of a single delay during the life of the project.

Cost Estimation in Software Projects

Cost estimation is essential for project planning, resource allocation, and risk management. Existing models like COCOMO, Function Point Analysis, and expert judgement techniques allow for cost prediction, but they fail in situations that are more dynamic and distributed and typically have several uncertainties at once (Rashid et al., 2020). The difficulty is worse in global software development (GSD) environments, as this area includes time zones, cultural issues, and communication problems, which introduce additional uncertainty (Khan et al., 2021).

Newer techniques have concentrated on hybrid and machine learning-based estimation. Deep learning models, for example, have the capacity to learn nonlinear patterns across cost drivers, which offer a considerable advantage in terms of accuracy as compared to more traditional methods (Ul Hassan et al., 2022). However, most models remain inadequate in accounting for the domino effect associated with delays that CDDE directly addresses by correlating time overruns with incremental monetary costs.

Definition and Nature of Project Delays

A delay in a project is generally regarded as a departure from the scheduled timeline, usually resulting in wasted cost and/or reduced profit (Shui et al., 2023). Unlike construction projects, software development is highly iterative and flexible, rendering it particularly vulnerable to evolving requirements, scope creep, and poor time estimation (Binu, 2025). These conditions

exacerbate cumulative delays, where inefficiencies at early stages cascade into compounded bottlenecks in later phases. For example, inaccurate requirement analysis can lead to design flaws, which in turn extend testing cycles and rework costs.

Project delays in software development are well understood to be a serious factor with negative implications for software development, for example, increased costs, lower quality, and increasing disputes among the stakeholders, which all lead to project failure or a project with a troubled outcome. Project delays in software development can stem from several antecedents, especially project governance (PG) and information technology governance (ITG), which determine decision-making, fault, responsibility, and supervision throughout project execution. In addition, there are significantly negative relations with ITG predicting project delay (PD) and PG predicting PD, to support both hypotheses. The results imply that stronger governance mechanisms related to the area of project management and the information technology governance theme that are aligned with a specific project will help to manage PD. Weak governance structures related to project management and information technology governance themes will prompt delays and highlight the role that organizational policies, leadership, and information technology alignment have to play in ensuring an organization can deliver software projects on time and within time (Khalid et al., 2022).

The numerous negative and complex factors resulting in software project failure are often linked to drift from remnant schedules. Delay is one of the most significant root causes of software project failure, and it is critical to find ways to detect delay signals sooner rather than later in order to achieve the ultimate goal of quality in software project development and to limit risks of budget overruns and quality issues. One potentially useful avenue of addressing the complexity of projects is the use of predictive models, and in particular models that use machine learning, which have seen good success in other domains. In this study, a novel AI-based approach using an explainable graph neural network (GNN) has been proposed to predict delays in software projects. The GNN approach realizes improved predictive abilities to predict delays and also provides interpretable predictions about where delays may occur. The experimental results show that the model predicts delays with an improved average of 4% accuracy, provides stability as it predicts stability in tasks across iterations, and provides meaningful explanations for predicting delays at the optimal threshold of classification. Such explainable AI models are a significant leap forward in project management and provide managers with reliable predictions about tasks and interpretations that can assist in risk mitigation, thus decreasing the probability of project failure due to schedule slippage (Tomura & Dam, 2024).

The additive nature of delays shows the importance of analytic models for dynamic delays rather than just static time slippage. CDDE provides the ability to analyze the additive and multiplicative effects of delays across flow phases and can provide a much more realistic basis for cost forecasts.

Effects of Delays on Cost Estimation

Delays compromise the accuracy of cost estimates as they alter scheduled resource allocations and require re-budgeting. Traditional models rarely include consequential effects of delays, which will be absorbed by costs like hidden overhead, increased labor cost, and wasted resources (Ahmed et al., 2021). For instance, if integration testing occurs late in a project's life cycle, the project team may need to pay overtime labor costs, increase defect rates, and push product delivery - all of which create a financial loss.

Research calls for adaptable models that utilize up-to-date delay information, instead of determined, static estimates (Chiang & Lin, 2020). CDDE's contribution is the ability to quantify the cumulative duration of delays and make the connections with financial impacts, thus linking schedule monitoring to cost monitoring.

Role of GQM and Machine Learning in Delay Analysis

The Goal-Question-Metric (GQM) method offers a systematic process for translating an organization goal into quantifiable observable indicators in a project. For delays it provides managers with a way to articulate goals (i.e., minimize schedule). The GQM method allows managers to produce questions for practical understanding, but importantly breaks down into metrics (i.e., number of delayed tasks, average slippage duration) (Zhao et al., 2023). GQM extends CDDE by linking delays to measurable entities that relate to project-defined goals.

Machine learning can also enhance estimation. Algorithms such as support vector machines, decision trees, and deep learning networks are effective at uncovering non-linear relationships between project variables (Taboada et al., 2023). Their predictions are often still "black box"; GQM provides a usable mathematical relationship to link delays and costs, which can be interpreted by decision makers.

4. Theoretical Framework

- From a Systems Theory standpoint, projects operate as systems of interdependent systems, where any disturbance in a subsystem is felt systemically (Mamedova & Bolonin, 2025). CDDE formalizes this by examining the aggregate effects of small delays across phases.
- Contingency Theory highlights that there is no single estimation technique that works in all contexts; approaches must be adjusted to specific situational variables, such as project size, complexity, and risk (Banywesize, 2022). CDDE agrees with this flexibility by allowing for appropriate calibration within a project.
- Goal-Oriented Measurement (GQM) matches measurement to organizational goals and provides a structured pathway for linking objectives to measurable outcomes (Calabrese et al., 2020). CDDE can be incorporated into GQM methods to enable a systematic monitoring of indicators of delay-related outcomes.

All three perspectives support CDDE as a holistic, adaptable, and measurable framework for managing delays in software projects and their impact on costs.

5. Practical projects

These specific cases supported by literature and CDDE implementation will be explored as follows:

1. Health Sector projects

- **Electronic Health Records (EHR):** There is literature that highlights how EHR implementation delays usually occur, as organizations often underestimate the complexity of migrating data, which is often significant enough to prolong project timelines and consequently budgets. CDDE could quantify the impact delays in requirements from early phases have on delayed testing, and the resultant monetary implications to the organization's information technology budget (Rahikkala et al., 2016).
- **Telemedicine Platforms:** Subsequently, delays in integrating software for telehealth services resulted in service cessation during the COVID-19 pandemic. The CDDE application would highlight how interface testing delays multiply costs due to urgent rollout needs (Sepasgozar et al., 2019).
- **Pharmacy Information Systems:** Scope creep in pharmaceutical IT systems often causes delays in regulatory compliance, resulting in penalties. CDDE could model cumulative penalties and additional labor costs.
- **Hospital Logistics Software:** Delays in implementing patient logistics systems increase resource wastage. CDDE helps quantify overtime and misallocation costs.

2. Government Sector projects

- **National ID Projects:** Government identity management systems often face political and technical delays. CDDE can forecast cost escalations due to phased rollouts extending years beyond initial estimates (Aljedaibi & Alsulami, 2021).
- **E-Government Portals:** Digital service delays impact citizen trust. Using CDDE, managers can quantify cost increases tied to vendor disputes and scope creep.
- **Taxation Systems:** Software delays in tax portals cause revenue collection losses. The CDDE application demonstrates how missed deadlines compound into financial shortfalls.
- **Smart City Initiatives:** IoT-enabled governance platforms often face cumulative delays due to vendor integration challenges. CDDE provides metrics for cost and schedule recalibration.

3. Education Sector projects

- **Learning Management Systems (LMS):** Universities deploying LMS platforms often experience integration delays with existing IT systems, leading to higher costs. CDDE quantifies compounding impacts of underestimated testing phases (Hundhausen et al., 2021).
- **Digital Curriculum Platforms:** Delays in digital textbook rollouts affect academic schedules. CDDE illustrates how iterative rework extends both time and financial investment.

- **MOOCs and E-Learning Platforms:** Massive online learning systems face scalability issues, where underestimated server integration delays cause exponential budget overruns.

4. Logistics Sector projects

- **Supply Chain Management Systems:** Delays in SCM software create cascading inefficiencies in procurement. CDDE demonstrates how early module delays inflate logistics costs.
- **Warehouse Automation:** Implementation delays in automated tracking systems increase storage and labor costs. CDDE highlights cumulative penalty costs.
- **Fleet Management Systems:** Transport projects delayed due to GPS and IoT integration, see exponential cost increases, quantifiable via CDDE.
- **Customs Clearance Systems:** National logistics portals delayed by integration challenges result in trade inefficiencies. CDDE provides structured cost forecasts for the delay duration.

5. Cdde Model And Equation

To recreate multidimensional delays, the CDDE model is designed using five primary delay groups. The coefficient for each type of delay is multiplied by a corresponding weight that reflects the severity of its impact on the project, as follows:

$$CDDE = \left(\frac{W_s \times D_s + W_h \times D_h + W_m \times D_m + W_e \times D_e + W_r \times D_r}{T_p} \right) \times 100$$

This formula provides a percentage value, referred to as the cumulative delay percentage, where:

D_s : Delays due to scheduling (e.g., overoptimistic schedule).

D_h : Human resource delays (due to a lack of skilled staff).

D_m : Delays caused by administrative or management issues (such as permits).

D_e : Delays due to exposure factors (e.g., suppliers and force majeure).

D_r : Delays resulting from the absence of resources (hardware/software/tools).

W_x : Weight assigned to each delay type, reflecting its contribution.

T_p : Total planned project duration (days)

All delay durations (D_x) are then weighted (W_x) based on their respective significance within the project framework. The total of all weighted delays (TWD) is normalized by dividing it by the total duration of the planned project (T_p) and multiplying by 100 to obtain a percentage.

The proposed formula was applied to over 52 projects and consistently demonstrated high levels of accuracy and reliability. Selected case studies are presented herein to illustrate its effectiveness.

6. Methodology

A combination of methods, including interviews with experts, empirical data from surveys, and historical project documentation, was employed in this study. The weight was determined using the Delphi method. CDDE was computed using an Excel-based model.

We combined qualitative and quantitative methods to ensure that the CDDE model was robust. We identified four projects that experienced identifiable delays in healthcare and analyzed them retrospectively based on project documents and stakeholder interviews.

The Delphi process was used to assign weights to the different types of delays. Experts were asked to rate these types of delays and their impacts using a five-point scale. Subsequently, these ratings were averaged and normalized. Surveys and interviews were conducted with project managers, and their teams led the data collection. Records from project logs were analyzed to determine the duration of the delays.

Calculations were performed in Excel, and the results were cross-validated using stakeholder feedback.

7. Comparative Analysis With Established Models

Table 2 provides a comparative analysis of three established project evaluation models—COCOMO, EVM, and the proposed CDDE. Although the comparison is contextualized within a surveillance system deployment, its primary aim is to highlight the key distinctions among the models in terms of their strengths, limitations, and effectiveness in attributing delays. This analysis emphasizes the unique diagnostic capabilities that the CDDE model offers in contrast to traditional evaluation approaches.

Table 2. Comparative analysis of COCOMO, EVM, and CDDE models in the context of a surveillance system deployment

Model	Strengths	Limitations	Delay Attribution Capability
COCOMO	Effort estimation based on code size	Not adaptive to real-time project dynamics	None (does not address delay sources)
EVM	Tracks progress via cost and schedule variance	Lacks specificity in diagnosing delay causes	Indirect (via schedule variance only)
CDDE	Identifies, quantifies, and weights delay types	Requires detailed stakeholder input and classification	Direct (categorized, weighted delays)

8. Case Studies

Four projects with real-world applications were analyzed: ERP implementation, surveillance camera systems, web portals, and Infant Protection System (IPS). All four projects were examined using the CDDE model to identify common disturbance patterns and their extent.

8.1. Case Study 1 – Erp Implementation

The ERP system was implemented in a hospital for a planned duration of 300 days. However, the project faced several obstacles and ultimately took 480 days to complete (Table 3).

The table below summarizes the categorized delays and their weights for the ERP implementation case study.

Table 3. Delay breakdown for the ERP implementation project

Delay Type	Duration (D)	Weight (W)	Weighted Delay (W×D)
Scheduling (D _s)	40	0.25	10.00
Human Resources (D _h)	60	0.30	18.00
Management (D _m)	70	0.25	17.50
External (D _e)	80	0.25	20.00
Resources (D _r)	55.05	0.30	16.51
			Total = 82.01

$$CDDE = (82.01 / 300) \times 100 = 27.34\%$$

8.2. Case Study 2 – Surveillance Camera System

This security project had a planned timeline of 180 days but lasted 360 days in reality. The primary causes of delay were vendor-related issues and resource availability (Table 4).

Table 4. Delay breakdown and weight assignments for the Surveillance Camera System case study

Delay Type	Delay Duration (D)	Weight (W)	Weighted Delay (W × D)
Scheduling (D _s)	20 days	0.15	3.00
Human Resources (D _h)	30 days	0.2	6.00
Management (D _m)	40 days	0.2	8.00
External (D _e)	60 days	0.3	18.00
Resources (D _r)	30 days	0.15	4.50
			Total = 39.5

$$CDDE = (39.5 / 180) \times 100 = 21.94\%$$

8.3. Case Study 3 – Web Portal

This project had a planned timeline of 300 days but took 449 days to complete. The main reasons for the delay were vendor-related issues and resource availability (Table 5).

Table 5. Delay breakdown and comparative analysis for the Web Portal case study

Delay Type	Duration (D)	Weight (W)	Weighted Delay (W×D)
Scheduling (D_s)	25	0.20	5.00
Human Resources (D_h)	30	0.30	9.00
Management (D_m)	20	0.25	5.00
External (D_e)	40	0.25	10.00
Resources (D_r)	34	0.25	8.50
			Total = 37.5

$$CDDE = (37.5 / 300) \times 100 = 12.50\%$$

8.4. Case Study 4 – Infant Protection System

This Infant Protection System had a planned timeline of 360 days but ultimately lasted 730 days. The primary causes of delay were management and external factors (Table 6).

Table 6. Case Study 4 – Infant Protection System

Delay Type	Delay Duration (D)	Weight (W)	Weighted Delay (W × D)
Scheduling (D_s)	60 days	0.20	12.00
Human Resources (D_h)	70 days	0.30	21.00
Management (D_m)	80 days	0.25	20.00
External (D_e)	100 days	0.25	25.00
Resources (D_r)	60 days	0.20	12.00
			Total = 90.0

$$CDDE = (90.0 / 360) \times 100 = 25.00\%$$

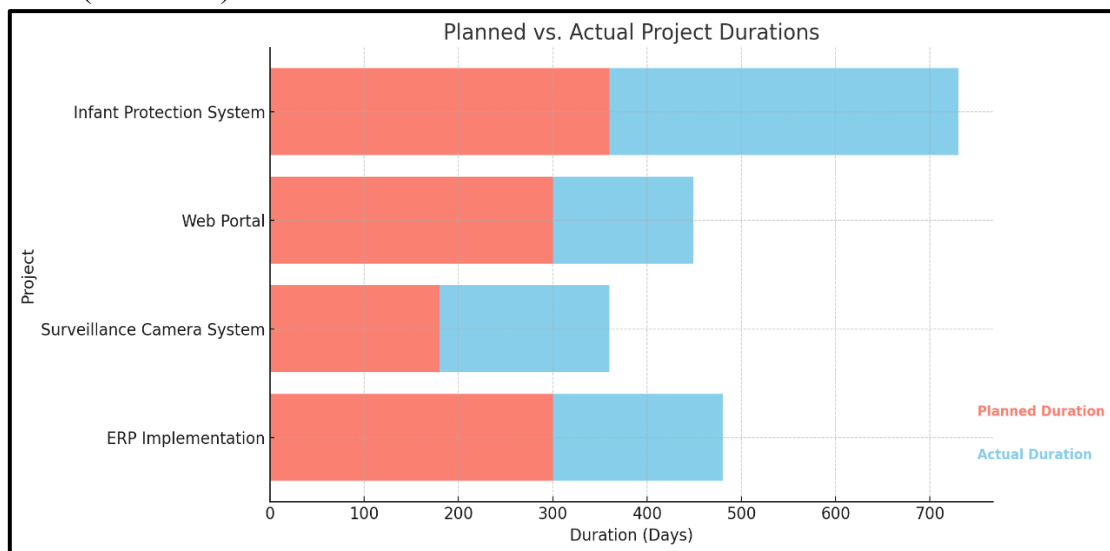


Figure (1): Planned vs. Actual Durations of the Four Case Study Projects

9. Quantitative Evaluation of CDDE Accuracy

The results of the Cumulative Delay Duration Equation (CDDE) were contrasted with the actual delay percentages obtained from the discrepancy between the planned and actual project durations in order to evaluate the numerical accuracy of the CDDE. The coefficient of determination (R^2), mean absolute error (MAE), and root mean square error (RMSE)—three commonly used accuracy indicators—were used. The comparison of the four case studies' actual and CDDE-based delay percentages is shown in Table (X).

Table 7. Comparison between Actual Delays and CDDE-based Delay Percentages

Project	Planned Duration (days)	Actual Duration (days)	Actual Delay (%)	CDDE (%)
ERP	300	480	60.0	27.34
Cameras	180	360	100.0	21.94
Web Portal	300	449	49.7	12.50
Infant Protection	360	730	102.8	25.00

Significant differences between the CDDE values and the actual delay percentages were found by the quantitative analysis. With an R^2 of 0.21, MAE of 56.42%, and RMSE of 60.40%, the statistical indicators support this finding. These findings imply that reproducing the exact discrepancy between planned and actual durations is not the goal of the CDDE. Rather than measuring the absolute time slippage, it serves as a diagnostic and analytical model that captures the weighted influence of various delay categories (such as management, resources, and external factors). Therefore, even though CDDE might not be an accurate indicator of overall delay percentages, it offers important information about how delay causes are distributed, weighted, and prioritized, which increases its usefulness in project risk mitigation and managerial decision-making.

10. Analysis And Discussion

The CDDE values indicate variations in the severity of delays. Projects with high CDDE values are significantly correlated with cost overruns. This model provides a realistic and detailed view of delays, helping project managers identify the reasons for the delays and plan for mitigation.

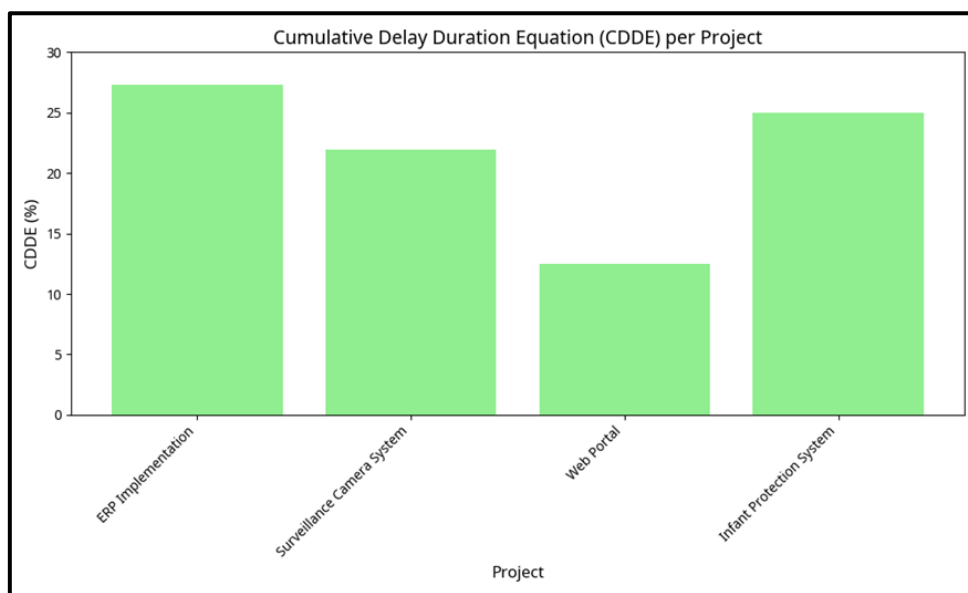


Figure (2): Cumulative Delay Duration Equation (CDDE) Values for the Case Study Projects

Budget overruns are strongly related to high values of CDDE. The largest CDDE of 27.17% was documented in the ERP implementation because the hardware delivery was delayed significantly and there were resource deficits. This shows that the model represents delay-causal chains- between root causes and the total impacts.

The mean average CDDE was approximately 21.65%, which forms a realistic baseline with which to compare them in the future. Enterprises can establish a point of reference such as CDDE by 21 per cent to hint at issues. Reviewing the weights of particular delays, the managers can focus on which types of delays are the most significant to improve. Outside the world of academia, there are obvious industry advantages of the CDDE model. Its products allow you to predict early delays, negotiate with suppliers by providing quantitative information of the financial effect of slippages, and optimise resource utilisation based on weighted delay factors. These applications are demonstrations of how the model is used as a decision-support tool that facilitates the connexion between theory and the real-life project management.

11. Correlation Analysis Between CDDE and CIDE

To better understand the statistical relationship between project delay (CDDE) and cost impact (CIDE), we calculated both the Pearson and Spearman correlation coefficients.

11.1. Pearson Correlation

However, the corresponding p-value of 0.2150 exceeds the conventional significance threshold of 0.05, indicating that the result is not statistically significant. As such, this relationship should be interpreted with caution.

11.2. Spearman Correlation

The Spearman rank correlation coefficient was $\rho = 0.800$, suggesting a strong positive monotonic association between CDDE and CIDE. Despite the strength of this numerical relationship, the p-value of 0.2000 surpasses the standard threshold of 0.05, indicating a lack of statistical significance. Therefore, although the observed correlation may suggest a potential link, it does not constitute conclusive evidence and should be interpreted with appropriate caution.

11.3 Root Cause Analysis of Delays

A root cause analysis was conducted with the help of Pareto principle (80/20 rule) to get a closer idea of how projects are delayed. The theory postulates that a few factors create most problems. This approach determines and prioritises the delay types that have the highest contributions to total delays in all studies. The Pareto chart below illustrates how these causes are distributed and which of them should be solved first in order to enhance project performance to the greatest extent possible.

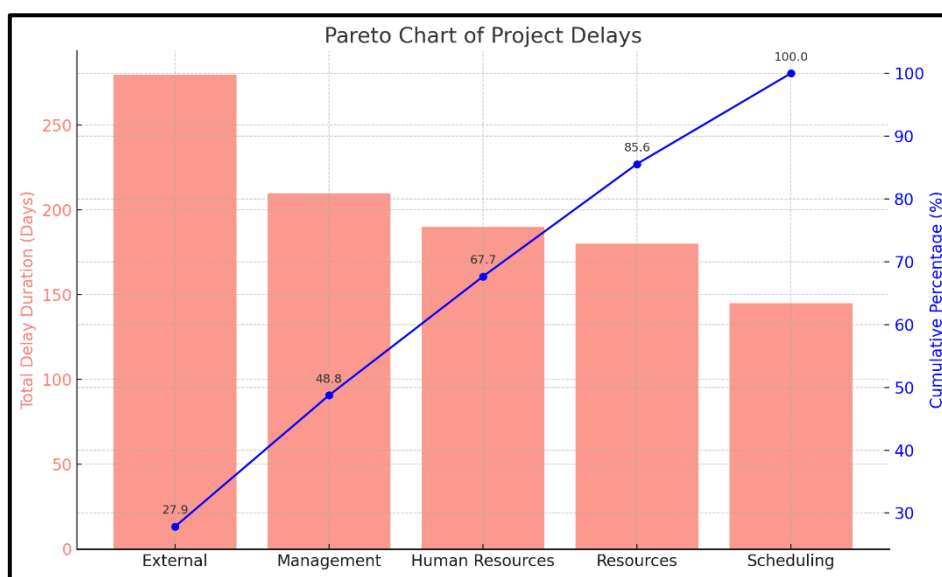


Figure (3): Pareto Chart of the Principal Causes of Project Delays

According to the Pareto analysis, a small number of crucial factors account for the majority of delays. In particular, the majority of cumulative delays are caused by management-related problems and resource shortages, which are followed by external dependencies like vendor performance and regulatory approvals. On the other hand, although they are still present, scheduling errors and human resource limitations have a relatively smaller overall impact on delays. Given that project overruns could be considerably decreased by addressing a small number of high-impact causes, this finding supports the 80/20 principle. Therefore, in order to achieve more lasting and effective improvements, project managers should give priority to mitigation strategies that target these major delay categories.

12. Validation And Evidence

The Delphi panel was comprised of ten senior software-project management gurus with a track record in large-scale enterprise and healthcare IT experience. Their practical experience in handling delivery risks, analysing the causes of delays and nourishing strategic project analysis gave them a comprehensive, informed weighting of the CDDE model. The CDDE model was tested on four case studies belonging to real software projects- the case studies test its relevance. Findings indicated a strong correlation between the CDDE ratio, cost overrun and schedule delays. ERP and surveillance-camera projects found that high CDDE values are strongly associated with budget overruns, resource hoarding, and inefficiencies found in management audits. There are major industry gains. PMO employee interviews demonstrate that the model is a supplement to traditional planning and helps post-project reviews and mitigation planning to prevent future slippage. It also facilitates the use of evidence-based decisions, since it offers a structured direction on the procurement of resources, the allocation of resources, and the ranking of high-impact delay types with a weighted analysis.

13. Cost Impact Of Delay Equation (CIDE)

While finance and business leaders recognize that a project may experience delays, which may have serious cost implications, they often lack the clarity and precision required to make informed decisions.

CIDE is a practical model that transforms cumulative delay percentage (CDDE) into an accurate financial cost estimate. It involves multiplying the scheduled project duration by the average daily cost, producing a figure that reflects the financial repercussions of delays.

The equation is expressed as follows:

$$CIDE = \left(\frac{CDDE}{100} \times T_p \right) \times C_d$$

where:

CDDE is the delay distribution extracted from the delay classification model, T_p denotes the planned duration of the project (in days), C_d represents the average cost per project day, and CIDE represents the total estimated cost of delay.

By allowing project managers (PMs), financial analysts (FAs), and stakeholders to move away from abstract delay metrics, this approach translates delays into financial terms that stakeholders can understand. It aids in calculating budget overruns and prepares teams for contractual discussions and risk mitigation strategies.

Example case:

For instance, if a project was scheduled for 300 days, the average daily cost was forecasted to be USD 1,000, and the CDDE was calculated at 27%, the financial cost of delay would be:

$$CIDE = \left(\frac{27}{100} \times 300 \right) \times 1000 = 81,000 \text{ USD}$$

Based on this scenario, the delay would incur an additional cost of USD 81,000, a figure that can be seamlessly integrated into financial analyses to support timely corrective actions.

14. Statistical Analysis: CDDE Vs. CIDE

This section presents the relationship between CDDE and CIDE for four real-world software projects. The financial impact of the delay was computed using the following formula:

$$CIDE = ((CDDE / 100) \times \text{Planned Duration}) \times \text{Daily Cost}.$$

A strong positive correlation (Pearson’s $r = 0.79$) was observed, indicating that higher delay percentages tended to result in greater financial costs, thereby underscoring the practical value of using the CDDE as an early indicator of budget risk.

As shown in Figure 2, ERP and Infant Protection System projects demonstrate relatively high CDDE percentages, corresponding to greater financial losses due to delays. In contrast, the Web Portal project exhibits a lower CDDE and a correspondingly lower cost impact. The Cameras project shows moderate values for both CDDE and CIDE, placing it between the high-impact and low-impact cases.

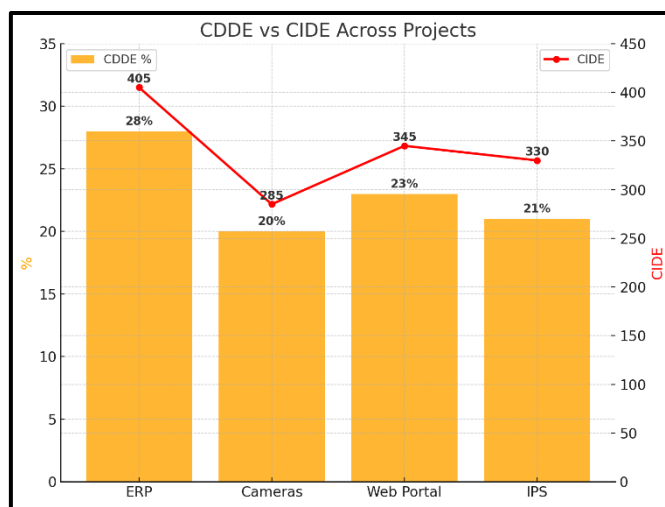


Figure 4. CDDE versus CIDE for four software projects

15. Conclusion

To improve the quantification and understanding of software project delays, the CDDE model integrates the GQM framework with empirically derived weights based on expert input. This structured, data-driven method analyzes delay impacts, moving beyond surface-level observations to provide a deeper perspective on the root causes of delays, whether they arise from human factors, scheduling gaps, management inefficiencies, or external disruptions.

Applying the CDDE model to real-world software projects, particularly in healthcare IT environments such as ERP implementations, consistently demonstrates a clear correlation between categorized delays and overall project performance. Notably, the inclusion of CIDE

adds a critical financial dimension to the analysis by translating abstract, time-based delay data into concrete cost implications.

The novelty of this study lies in its effort to unify the delay assessment process using a flexible and extensible framework. Unlike generalized models such as COCOMO or EVM, which depend on predefined estimation or progress tracking formulas, the CDDE model adapts to the unique challenges of each project through stakeholder-driven weighting and domain-specific categorization of delay types.

Besides being innovative, the CDDE model has a high potential of worldwide applicability. Its design is not geographically or industry-specifically based although it was designed and tested in Saudi Arabia when it was first developed and tested on software projects. The basis of the model on the GQM framework provides the ability to recalibrate the model through expert judgement in any organisational or cultural setting. Its versatility allows it to be used not only in the assessment of delays in healthcare systems but also in the assessment of delays in the works of the public sector, education, logistics, defence systems, and software projects in enterprises all over the world. With organisations in all regions demanding more trusted and context-driven delay analytics, the CDDE model offers a promising universal method to measure, analyse, and address project delays accurately and in a realistic manner.

In general, the model is a useful contribution to software project management as it addresses the gap between theoretical models of delay and the reality that is complicated by live implementation. It aids data-driven decision making, post-project assessments, and gives stakeholders a clear, flexible instrument on future planning and project control.

Despite the promising outcomes of CDDE and CIDE models in measuring delays in software projects and the cost consequences, there are a number of limitations that should be recognised. First, delay weights are calculated with a lot of expert judgement using the Delphi method that, though strong, can bring the subjectivity depending on the backgrounds or prejudices of participants involved. Also, the model has been confirmed in very few case studies in enterprise level software projects especially in the health care sector. Nevertheless, whether it can be generalised to other areas, including startups or highly agile settings, where delay dynamics are much different, is yet to be examined.

The CDDE model's limited scope may limit its generalizability, despite the fact that it was validated using healthcare IT projects. Projects from other domains, like government IT, logistics, and education, should undergo preliminary testing to improve robustness. The flexibility and adaptability of the CDDE framework would be demonstrated by such cross-domain validation, enhancing its potential as a universal delay diagnostic tool.

Another limitation lies in the manual nature of data collection and classification of delays. Automating this process through project management tools or integration with real-time tracking systems could enhance the model's scalability and reliability.

Further research is encouraged to enhance the weight assignment process through data-driven methods, including machine learning techniques capable of dynamically identifying delay patterns across diverse project contexts. Moreover, expanding the validation scope to include a wider range of industries and international projects would strengthen the model's

robustness. Integrating CDDE into real-time dashboards for proactive delay monitoring is another valuable direction for practical deployment.

Declaration Of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Declaration Of Generative Ai In Scientific Writing

No generative AI or AI-assisted technologies were used in the writing of this manuscript.

Data Availability

The data supporting the findings of this study are not publicly available due to institutional confidentiality agreements. However, non-sensitive data or summary information may be made available from the corresponding author upon reasonable request.

Author Contributions

Eng. Ahmad Abdullah Alghamdi: Conceptualization, Data curation, Formal analysis, Investigation, Methodology, Software, Visualization, Writing – original draft. **Prof. Abdullah Al-Qarni:** Conceptualization, Formal analysis, Funding acquisition, Project administration, Resources, Supervision, Writing – review & editing. **Dr. Wajdi Al-Judaibi:** Conceptualization, Formal analysis, Methodology, Supervision, Validation, Writing – review & editing.

Acknowledgment

The authors would like to extend their heartfelt thanks to the College of Computer Science at King Abdulaziz University for its ongoing support and academic encouragement throughout the course of this research. Sincere appreciation is also due to the project managers and subject matter experts who participated in the surveys and interviews. Their thoughtful input and practical perspectives played a crucial role in shaping the empirical findings of this study.

Funding

This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

Ethical Considerations

This study was conducted in accordance with the ethical standards of academic research involving institutional data. Formal approval to access and use project documentation, interviews, and performance metrics was obtained from the Armed Forces Hospital Administration of the Kingdom of Saudi Arabia. The data used in this study were collected exclusively for research purposes, with all necessary permissions obtained prior to analysis. No personal or sensitive information was disclosed, and confidentiality was upheld. The authors confirmed that the use of all project-related data adhered to institutional policies and

ethical guidelines, and informed consent was obtained from participants involved in interviews or survey-based data collection.

Abbreviations And Notations

Symbol / Term	Description
CDDE	Cumulative delay duration equation – A percentage-based metric quantifying the overall impact of multiple types of delays in software projects.
CIDE	Cost impact of delay equation – A monetary estimation model derived from CDDE and project financial parameters.
D_s	Delay due to scheduling – Refers to project delays caused by unrealistic timelines or misaligned scheduling.
D_h	Delay due to human resources – Includes staffing shortages, skill mismatches, or team turnover.
D_m	Delay due to management – Delays arising from decision-making bottlenecks, administrative issues, or internal approvals.
D_e	Delay due to external factors – Includes supplier issues, governmental approvals, or unforeseen external events.
D_r	Delay due to resource deficiencies – Unavailability of hardware, software, tools, or infrastructure.
W_s, W_h, W_m, W_e, W_r	Weights assigned to each delay type – These reflect the relative significance or severity of each delay category as determined by expert judgment.
T_p	Planned project duration – The original intended duration of the project in days.
C_d	Average daily project cost – The expected cost per day of the project execution.
TWD	Total weighted delay – The sum of all delays after being adjusted by their assigned weights.

References

- [1] Ahmed, M., Siddiqui, A. A., Khan, S., & Junaid, M. (2021). Software cost estimation in the global software development business: A review of models and cost drivers for economic business. *International Journal of Business and Economic Affairs*, 6(3), 118-129.
- [2] Aljedaibi, W., & Alsulami, A. A. (2021). A CMMI-Based Method for Software Development Process Assessment: Applying CMMI Process in Saudi

- Arabia. *International Transaction Journal of Engineering, Management, & Applied Sciences & Technologies*, 12(4), 1-10.
- [3] Banywesize, R. B. (2022). *Project length and management control*. Master. Kampala International University, Kampala.
- [4] Binu, A. (2025). *Delays in Software Projects: Identifying Contributing Factors*. Master. Jönköping university, Jönköping.
- [5] Calabrese, J., Esponda, S., Pasini, A., & Pesado, P. (2020, October). Data Evaluation Model Using GQM Approach. In *Argentine Congress of Computer Science*. Cham: Springer International Publishing, 141-154.
- [6] Chiang, H. Y., & Lin, B. M. (2020). A decision model for human resource allocation in project management of software development. *IEEE access*, 8, 38073-38081.
- [7] De Sousa, A. L., De Souza, C. R., & Reis, R. Q. (2022). A 20-year mapping of Bayesian belief networks in software project management. *IET Software*, 16(1), 14-28.
- [8] Hundhausen, C., Carter, A., Conrad, P., Tariq, A., & Adesope, O. (2021, March). Evaluating commit, issue and product quality in team software development projects. In *Proceedings of the 52nd ACM Technical Symposium on Computer Science Education*, 108-114.
- [9] Khalid, M., Khan, R. A., Khushnood, M., Aslam, S., Khattak, Z. Z., & Abbas, S. (2022). An Empirical analysis of the influence of project governance and information technology governance on project delay. *Global Business Review*, 1-20.
- [10] Khan, J. A., Khan, S. U. R., Iqbal, J., & Rehman, I. U. (2021). Empirical investigation about the factors affecting the cost estimation in global software development context. *IEEE Access*, 9, 22274-22294.
- [11] Koulinas, G. K., Xanthopoulos, A. S., Tsilipiras, T. T., & Koulouriotis, D. E. (2020). Schedule delay risk analysis in construction projects with a simulation-based expert system. *Buildings*, 10(8), 1-19.
- [12] Kula, E., Greuter, E., van Deursen, A., & Gousios, G. (2023, November). Dynamic prediction of delays in software projects using delay patterns and bayesian modeling. In *Proceedings of the 31st ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, 1012-1023.
- [13] Kuuttila, M., Mäntylä, M., Farooq, U., & Claes, M. (2020). Time pressure in software engineering: A systematic review. *Information and Software Technology*, 121, 1-70.
- [14] Mahdi, M. N., Mohamed Zabil, M. H., Ahmad, A. R., Ismail, R., Yusoff, Y., Cheng, L. K., ... & Happala Naidu, H. (2021). Software project management using machine learning technique—a review. *Applied Sciences*, 11(11), 1-39.
- [15] Mahmood, Y., Kama, N., Azmi, A., Khan, A. S., & Ali, M. (2022). Software effort estimation accuracy prediction of machine learning techniques: A systematic performance evaluation. *Software: Practice and experience*, 52(1), 39-65.
- [16] Mamedova, N., & Bolonin, T. (2025). Application of Systems Theory in the Design of System Monitoring Service of Hardware-Software Complex. *WSEAS Transactions on Systems*, 24, 288-302.

- [17] Rahikkala, J., Hyrynsalmi, S., Seppänen, M., & Leppänen, V. (2016, June). The impact of a delayed software project on product launch coordination: A case study. In *2016 International Conference on Engineering, Technology and Innovation/IEEE International Technology Management Conference (ICE/ITMC)*, 1-8. IEEE.
- [18] Sepasgozar, S. M., Karimi, R., Shirowzhan, S., Mojtahedi, M., Ebrahimzadeh, S., & McCarthy, D. (2019). Delay causes and emerging digital tools: A novel model of delay analysis, including integrated project delivery and PMBOK. *Buildings*, 9(9), 1-37.
- [19] Shui, S., Rifai, A. I., & Pamadi, M. (2023). The analysis performance of project delays in pile foundation work. *IJEBD (International Journal of Entrepreneurship and Business Development)*, 6(1), 172-182.
- [20] Taboada, I., Daneshpajouh, A., Toledo, N., & De Vass, T. (2023). Artificial intelligence enabled project management: a systematic literature review. *Applied Sciences*, 13(8), 1-23.
- [21] Tomura, S., & Dam, H. K. (2024). Generating explanations for AI-powered delay prediction in software projects. In *Generative AI for Effective Software Development*. Cham: Springer Nature Switzerland, 297-316.
- [22] Ul Hassan, C. A., Khan, M. S., Irfan, R., Iqbal, J., Hussain, S., Sajid Ullah, S., ... & Umar, F. (2022). Optimizing Deep Learning Model for Software Cost Estimation Using Hybrid Meta-Heuristic Algorithmic Approach. *Computational Intelligence and Neuroscience*, 2022(1), 1-20.
- [23] Zhao, L., Wu, Y., Wang, X., & Zhu, M. (2023). Using GQIM Model and Gompertz Model to Measure, Analyze, and Optimize the Whole Process of Software Project. *Mathematical Problems in Engineering*, 2023(1), 1-15.