

American International University-Bangladesh (AIUB)

**Time-Series Data in Critical path analysis for project management**

***Improving Project Forecasting Accuracy***

KAZI MARUF AHMAD ALIF (19-40044-1)

ARIFIN SABID (19-41513-3)

MD. FAHIM MONTASIR MOJUMDER (19-41630-3)

KAZI EMADUZZAMAN GELANI (19-41678-3)

*A Thesis submitted for the degree of Bachelor of Science(BSc) in Computer Science and Engineering (CSE) at*

*American International University Bangladesh in January,2024*

Faculty of Science and Technology (FST)

## Abstract

This thesis delves into the integration of Time-Series Data Analysis within the framework of Critical Path Analysis (CPA) in software project management. Employing Python and the NetworkX library, a model utilizing directed graphs is developed to illustrate interdependencies among project tasks. [1]The primary objective is to compute crucial metrics—Early Start (ES), Early Finish (EF), Late Start (LS), and Late Finish (LF) times for individual tasks, facilitating the identification of the critical path. The long-term goals of increasing the quality of service provided complex systems while reducing development risks, costs and time [2] . The study emphasizes the pivotal role of visual representations in comprehending critical paths within software project management. This approach may be used to determine the project's crucial tasks [3]. Methodologies for deriving early and late times are elucidated, highlighting the practical implications of identifying critical paths in optimizing project scheduling and resource allocation [4]. To improve project forecasting accuracy, a comprehensive methodology involving the collection and analysis of time-series data from software projects was implemented. Statistical modeling [5]and predictive algorithms were applied to identify patterns and trends in project timelines, [6] Critical Path Analysis flow diagrams and leading to more precise forecasting of project durations and potential bottlenecks. By leveraging time-series data in critical path analysis, this research holds significant potential to enhance project planning and decision-making in software development[7]. It provides a means for project managers to gain a better understanding of possible project delays and identify critical tasks requiring attention, resulting in improved resource allocation, risk mitigation, and heightened project success. Key findings highlight that integrating time-series data into critical path analysis significantly improves project forecasting accuracy. By considering historical project data, patterns were leveraged to make informed predictions, not only enhancing schedule accuracy but also providing insights into potential risks and dependencies. The implications of these findings hold substantial promise for the software industry, allowing organizations to better allocate resources, manage project timelines more efficiently, and minimize delays. This research opens opportunities for more efficient project planning, optimized resource allocation, and increased overall project success rates most serious problems [8]. In conclusion, this thesis contributes a comprehensive overview and insights into improving project forecasting accuracy through the integration of time-series data into critical path analysis for software project management. By addressing key questions related to methodology, significance, results, and implications, this research holds the potential to advance project management practice**.**

## Approval

The thesis titled **“Time-Series Data in Critical path analysis for project management”** has been submitted to the following respected members of the board of examiners of the department of computer science in partial fulfilment of the requirements for the degree of Bachelor of Science in Computer Science on **(January 21, 2024)** and has been accepted as satisfactory.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Supta Richard Philip  Lecturer Faculty of Science & Technology  American International University-Bangladesh | |  | Dr. Mohammad Mahmudul Hasan  Associate Professor Faculty of Science & Technology American International University-Bangladesh | |
|  | | | | |
|  | DR. AKINUL ISLAM JONY  Associate Professor, Head (UG)  Faculty of Science & Technology  American International University-Bangladesh | | |  |
| Prof. Dr. Dip Nandi  Associate Dean  Faculty of Science and Technology  American International University-Bangladesh | |  | Mashiour Rahman  Sr. Associate Professor & Dean-in-charge  Faculty of Science and Technology  American International University-Bangladesh | |

## Contributions by authors to the thesis

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **Kazi Maruf Ahmad Alif** | **Arifin Sabid** | **Md. Fahim Montasir Mojumder** | **Kazi Emaduzzaman Gelani** | **Contribution (%)** |
|  | *19-40044-1* | *19-41513-3* | *19-41630-3* | *19-41678-3* |
| Conceptualization | 25% | 25% | 25% | 25% | 100 % |
| Data curation | 25% | 25% | 25% | 25% | 100 % |
| Formal analysis | 25% | 25% | 25% | 25% | 100 % |
| Investigation | 25% | 25% | 25% | 25% | 100 % |
| Methodology | 25% | 25% | 25% | 25% | 100 % |
| Implementation | 25% | 25% | 25% | 25% | 100 % |
| Validation | 25% | 25% | 25% | 25% | 100 % |
| Theoretical derivations | 25% | 25% | 25% | 25% | 100 % |
| Preparation of figures | 25% | 25% | 25% | 25% | 100 % |
| Writing – original draft | 25% | 25% | 25% | 25% | 100 % |
| Writing – review & editing | 25% | 25% | 25% | 25% | 100 % |

## Acknowledgments

First of all, I would like to thank almighty Allah, for his grace in accomplishing our thesis timely and our families for giving us mental and financial support throughout the whole bachelors.

We would like to express our gratitude to the **Faculty of Science & Technology** to keep thesis credit in the curriculum of the graduation program and give us a scope of tasting the flavor or research work with our interest.

We also want to express our gratitude to our supervisor, **SUPTA RICHARD PHILIP** sir, who gave us motivation, guidance, and control for this venture. He has decent knowledge of programming advancement, and his personal circumstances, perpetual assistance made the task do able.

## Keywords

CPM, Critical Path, Project Management, Project Planning, Forecasting, Software Engineering, Time driven, Activity diagram,

# List of Figures

[Figure 1 : NetworkX's topological sort flowchart 21](#_Toc156609088)

[Figure 2 : Project Diagram 22](#_Toc156609089)

[Figure 3 : The Critical Path Diagram 23](#_Toc156609090)

# List of Tables

[Table 1. Table showing the earliest and latest time with their slack values 23](#_Toc156607948)

# List of Abbreviations and Symbols

Abbreviations

CPM Critical Path Method

CPA Critical Path Analysis

ET Earliest Time

LT Latest Time

WBS Work Breakdown Structure

Table of Contents

[Abstract 2](#_Toc156609294)

[Approval 3](#_Toc156609295)

[Contributions by authors to the thesis 4](#_Toc156609296)

[Acknowledgments 5](#_Toc156609297)

[Keywords 6](#_Toc156609298)

[List of Figures 7](#_Toc156609299)

[List of Tables 8](#_Toc156609300)

[List of Abbreviations and Symbols 9](#_Toc156609301)

[Introduction 11](#_Toc156609302)

[1.1 General Aims 11](#_Toc156609303)

[1.2 Specific Objectives 11](#_Toc156609304)

[1.3 Research Questions 12](#_Toc156609305)

[1.4 Overview 12](#_Toc156609306)

[1.5 Improving Project Forecasting Accuracy 13](#_Toc156609307)

[Literature review 14](#_Toc156609308)

[Methods 18](#_Toc156609309)

[3.1 Integrating Critical Path Method (CPM) in Software Project Management 19](#_Toc156609310)

[3.2 Reasoning for using NetworkX topology sort 20](#_Toc156609311)

[Results or findings 22](#_Toc156609312)

[4.1 Interpretation of Result 24](#_Toc156609313)

[Discussion 26](#_Toc156609314)

[Conclusion 28](#_Toc156609315)

[Bibliography 30](#_Toc156609316)

[Code 33](#_Toc156609317)

**Chapter 1**

# Introduction

In the world of software project management, understanding the complexities of temporal interdependencies stands as a pillar to the success of any project. how to integrate in fixed investment limited resources, through the lowest cost and the shortest time to complete the project management

Problem [9] . This study embarks on exploring the critical path analysis method, interwoven with time-series data analysis, to unearth the intricate relationships between tasks and their temporal interdependencies. These workflows can get pretty complicated because different processing parts need specific data in a certain order, and there's this pressure of getting everything done within tight deadlines. [10]. Drawing insights from recognized sources and established literature, in most of these models, capacity is depicted as a fixed limit on the total amount of a resource that can be used within a specific time period. [11]. This research aims to deepen our comprehension of time-driven project management practices.

Software project management requires planning, execution, and risk assessment also design by giving solutions to frequently occurring problems [12] .The accuracy of project timeline forecasting is pivotal for successful project completion. Nonetheless, forecasting inaccuracies often lead to resource wastage and project delays [13]. Understanding this problem's background is crucial, considering the unpredictability inherent in software projects, necessitating methodologies to enhance forecasting accuracy.

## General Aims

This research endeavors to bridge the existing gap by integrating Time-Series Data Analysis into Critical Path Analysis for software project management. The primary motivation is to refine project planning, scheduling accuracy, and risk assessment strategies including reduce testing cost and errors [14]. The results show the effectiveness of the CPM in, planning, scheduling, and organizing, coordinating, managing, and controlling of project time and cost [15] .The study's broader theme is to enhance the understanding of temporal interdependencies within software project management.

## Specific Objectives

Central to this endeavor is the objective to address the gap in research that exists in the synthesis of methodologies, focusing on the temporal intricacies within software project management. CPM networks help in time and cost efficiency [16], [17] . A CPM model can be built based on training data for prediction, classification, identification and detection [18]. This study is positioned to offer unique insights into refining project forecasting accuracy, enabling more efficient resource allocation and successful project completion.

## Research Questions

The primary research question guiding this investigation is how the integration of Time-Series Data Analysis within Critical Path Analysis enhances forecasting accuracy in software project management. Additionally, the study aims to explore:

* How does the interconnection between Critical Path Analysis and Time-Series Data Analysis influence the accuracy of project scheduling?
* What is the impact of integrating Time-Series Data Analysis on risk assessment strategies within software project management?
* What are the distinct patterns and trends uncovered by integrating Time-Series Data Analysis into Critical Path Analysis for software project management?
* To what extent does the amalgamation of Time-Series Data Analysis and Critical Path Analysis impact risk assessment strategies within software project management?

## Overview

This thesis critically examines the intricacies of managing time-sensitive dependencies in software projects through a detailed analysis of the critical path method. Each section is tailored to provide comprehensive insights into different aspects, forming the foundation for understanding temporal dependencies in project management. This introduction lays the groundwork for subsequent chapters, highlighting the significance of comprehending time-driven dependencies in software project management. Our Aim for the project is to develop a comprehensive framework that seamlessly integrates time-series data analysis techniques with Critical Path Analysis tailored to the unique context of software project management. Evaluate the impact of data quality and granularity on project forecasting accuracy when employing time-series data analysis within the framework of Critical Path Analysis.

Moreover, when we combine different elements, we can identify consistent patterns or trends that impact project completion time. For example, if some tasks are always late at certain times, we can plan to fix the problem and keep the project on track. By looking at past information from similar projects, managers can guess what might happen in the future. This helps them set realistic deadlines and decide how to use resources, which makes it more likely that they can finish projects on time and within budget.

## Improving Project Forecasting Accuracy

The integration of time-series data into critical path analysis for software project management is a pivotal strategy to enhance project forecasting accuracy. By amalgamating time-series data analysis into the critical path analysis framework, project managers gain valuable insights into project performance and progress. This infusion results in more precise forecasting and improved decision-making throughout the project's duration. Tracking the duration and completion time of tasks within software projects aids in identifying potential delays and bottlenecks, while also offering a comprehensive understanding of the critical path - the essential sequence of tasks vital to meet project deadlines.

Moreover, the integration enables the identification of cyclical patterns or trends that influence project timelines. For instance, consistent delays in certain tasks during specific periods can be anticipated and addressed preemptively to avoid detrimental effects on the project schedule. By analyzing historical data from similar projects, managers can predict future project outcomes, facilitating the establishment of realistic deadlines and resource allocation, which significantly increases the likelihood of completing projects on time and within budget. The accuracy of project forecasting heavily relies on the quality and consistency of time-series data collection methods[19]. Regular updates and diligent monitoring throughout the project's lifecycle are crucial to maintaining the accuracy of forecasts[20].

Overall, this approach significantly bolsters project forecasting accuracy within software project management by empowering managers to make informed decisions based on task duration analysis, the identification of cyclical trends, and historical data interpretation. The focus on refining forecasting accuracy through the integration of time-series data analysis into the critical path analysis framework is designed to amplify project planning, risk assessment, and scheduling effectiveness within the software project management domain. By using the CPM time of project execution and the critical path of the project can be known so that the project can be completed on time[16].

**Chapter 2**

# Literature review

Previous research on time-series data analysis within Critical Path Analysis (CPA) to enhance project forecasting accuracy in the context of software project management. Effective project management is integral to the success of any endeavor, with advancements in the field continually shaping methodologies and strategies. This literature review delves into key works that explore time-driven dependencies, project planning, critical path analysis, and temporal interdependencies in software development.

One foundational reference in this field is the comprehensive guide on project management advancements, offering valuable insights into the ever-evolving landscape of methodologies [21]. This guide serves as a fundamental resource, encompassing various aspects of project management. It provides a holistic view of methodologies that have adapted to diverse project scopes, sizes, and complexities over time. Understanding the nuanced applications of project management methodologies in real-world scenarios contributes to a more comprehensive view of their effectiveness. For instance, it's essential to explore how these methodologies accommodate agile development approaches, given their increasing prevalence in software projects. The guide by [21] on time-driven dependencies in software development significantly contributes to understanding the dynamic nature of task interdependencies. It emphasizes the temporal aspect, which is crucial for project managers aiming for more accurate project planning and execution. To further enrich this perspective, specific case studies or examples where time-driven dependencies played a pivotal role in project success or failure can be explored. Real-world applications provide concrete illustrations of theoretical concepts and enhance the applicability of the findings. Examining the adaptability of time-driven approaches across various industry sectors could shed light on their versatility. A pivotal study by [15] underscores the significance of time as a driving factor in project planning, providing practical approaches for project managers to enhance planning processes and meet deadlines. [13] . Further exploration could delve into specific methodologies proposed by [15] and their applicability across different project types. Moreover, discussing challenges and limitations associated with these methodologies would provide a balanced perspective. For example, are there scenarios where time-centric planning approaches may face constraints or prove less effective? Understanding these limitations can guide project managers in selecting the most suitable approach based on project characteristics. The exploration of software project management dynamics by [22] sheds light on adaptive strategies and the evolving nature of project tasks. To expand on this, a deeper dive into specific adaptive strategies highlighted by [22], and how these strategies address the challenges posed by the rapidly changing landscape of software development, would be insightful. How do these strategies cater to emerging technologies and evolving client requirements? Understanding the practical implications of adaptive strategies can guide project managers in navigating uncertainties and complexities in a dynamically evolving industry. Notably, [16] delves into time-series analysis specifically applied to project management, addressing the temporal dimension and offering methodologies to analyze historical project data. Further exploration could involve discussing specific techniques employed in time-series analysis and their effectiveness in uncovering patterns and trends. For instance, machine learning algorithms have shown promise in enhancing the accuracy of time-series analysis. How do these advanced analytical tools contribute to the precision of forecasting? Examining the intersection of data analytics and project management tools can offer a more detailed understanding of the technological landscape. Critical path analysis, a fundamental tool in project management, is elucidated by [13] emphasizing its importance in identifying key tasks that determine project duration. To expand on this, providing examples of projects where critical path analysis played a pivotal role in successful project completion would illustrate its practical significance. Additionally, discussing variations or adaptations of critical path analysis in different industries or project environments would enrich the understanding of its versatility. For example, how does critical path analysis apply to large-scale infrastructure projects compared to agile software development projects? Examining the adaptability of this tool across diverse project types provides insights into its generalizability. The work by [11] exploring temporal interdependencies in software development, crucial for ensuring smooth execution, contributes to a nuanced understanding of these interdependencies. Further exploration could involve discussing specific challenges related to managing temporal interdependencies and proposing strategies to mitigate these challenges. Real-world examples where effective management of temporal interdependencies led to project success would enhance the practical relevance of the findings. Additionally, exploring how emerging trends like remote work impact temporal interdependencies and how project management strategies are evolving in response to these trends could be considered. While the comprehensive exploration of time-runs in project management techniques by [23] offers valuable insights, acknowledging the potential limitations or criticisms of these techniques would provide a more balanced perspective. Additionally, discussing alternative approaches or complementary methodologies that address the shortcomings of time-run techniques would contribute to a more comprehensive understanding of project management techniques. For instance, are there scenarios where time-run techniques may lead to suboptimal outcomes, exploring alternative methodologies or hybrid approaches ensures a more nuanced view of project management techniques. Despite the valuable insights provided by existing literature, it is essential to acknowledge gaps and limitations. The review mentions a lack of a comprehensive survey of existing methods and research in the field of software evaluation [23]. Further exploration could involve proposing potential areas of research or methodologies that could address this gap. Additionally, discussing the implications of this gap on the broader field of software project management research would contribute to a more holistic understanding.

In summary, the literature reviewed spans various aspects of project management, with a focus on time-driven approaches, critical path analysis, and temporal interdependencies in software development. These technologies offer new possibilities for automating certain concerns have been raised about their impact on traditional software development roles and practices [24]. The works collectively contribute to a foundation for the exploration of enhanced project forecasting accuracy through the integration of time-series data analysis with Critical Path Analysis in the dynamic landscape of software project management. The current research says that using time-series data analysis with Critical Path Analysis can make project forecasting more effectively emphasizes the growing importance [25]. However, there are still a few difficulties we face. We can see problems with how good the data is, the need for better tools to analyze it, and how complex software development projects are[26]. In simpler phrase terms, there is not enough research on how to use time-series data analysis together with Critical Path Analysis in the field of software project management. Effective software development projects heavily depend on the ability to forecast project progress accurately. It helps project managers make smart decisions about how to use resources, handle risks, and make sure the project timelines meet the expectations of everyone involved[27]. In recent years, people have become very interested in combining time-series data analysis techniques with Critical Path Analysis (CPA). [28]Critically evaluates the effectiveness of various computer software packages in teaching Engineering Management Science, focusing on the application of critical path methods [29]. This merging of different things is ready to go beyond traditional ways of predicting a project's outcome.

This literature review explores a fascinating journey through this ever-changing field, delving into previous research while also showing the way for future studies. The combination of analyzing data over time and using CPA (Certified Public Accountant) presents a great opportunity to improve the accuracy and effectiveness of managing software development projects. By carefully analyzing the existing literature and exploring the gaps in knowledge, this review aims to make a valuable contribution to the improvement of software project management and forecasting.

This extended literature review spans various aspects of project management, focusing on time-driven approaches, critical path analysis, and temporal interdependencies in software development. Each section has been further elaborated by exploring specific methodologies, providing additional examples, and discussing related research in more detail. By delving into the practical applications, challenges, and adaptability of these methodologies, this review offers to a comprehensive understanding of the current state of research in the integration of time-series data analysis with Critical Path Analysis for improved project forecasting accuracy in software project management.

**Chapter 3**

# Methods

The critical path method is a mathematically-based algorithm used for scheduling project activities and is a key component of effective project management. Any project involving interdependent activities can employ this method for scheduling. The process for utilizing the Critical Path Method (CPM) involves constructing a comprehensive project model, which includes:

* **List of Activities**: All activities required to complete the project, also known as the Work Breakdown Structure (WBS).
* **Activity Durations**: The estimated time needed for completion of each activity.
* **Activity Dependencies**: Defining the relationships and dependencies between the different project activities.
* **Longest Path Calculation**: Determining the longest sequence of planned activities essential for the project’s completion.
* **Earliest and Latest Activity Times**: Assigning earliest and latest times for each activity, ensuring a clear timeline for project execution.

This structured method, based on these elements, helps identify the critical path within a project. It allows project managers to focus on activities crucial to the project’s timeline and success.

**Critical Path Method Approach**

The Critical Path Method (CPM) is a pivotal, mathematically-based algorithm used to schedule project activities in an effective and organized manner within project management. It can be applied to any project containing interdependent activities.

* **Project Model Construction:**
  + Creation of a comprehensive project model that encompasses:
    - A breakdown of all activities required for project completion (also known as Work Breakdown Structure).
    - Time duration estimation for each activity.
    - Identification and definition of interdependencies among these activities.
* **Longest Path Calculation:**
  + Determination of the longest sequence of planned activities that culminate in the project's completion.
* **Key Terminology:**
  + Earliest Event Time (ETi): The earliest feasible occurrence time for the event associated with node (i).
  + Latest Event Time (LTi): The latest possible occurrence time for the event associated with node (i) without causing project delay.
  + Total Float (TFij) of activity (i, j): Represents the permissible delay in the starting time of activity (i, j) without impacting the project's completion time.
* **Critical Path Computation:**
  + Early Time Calculation: Determined by max [ETi, ETi + dij].
  + Late Time Calculation: Computed using min [LTi, LTi - dij].
  + Total Float Calculation: Derived as |ET - LT|.

We delve into the synergistic approach of merging Critical Path Analysis (CPA) with Time-Series Data Analysis to enrich software project management strategies. The objective is to elevate project planning, scheduling precision, and risk assessment within this domain. Our methodology heavily relies on relevant datasets, case studies, and scholarly articles in software project management. We meticulously extract and analyze Time-Series Data, employing techniques like data normalization, trend analysis, and integrating this data with Critical Path Identification. This thorough approach allows us to understand temporal dependencies and refine planning, risk mitigation, and resource allocation in software project management. The integration of Critical Path Analysis with Time-Series Data aims to unravel complexities in project planning. It stands as a significant milestone in addressing critical research questions and advancing the understanding of Time-Series Data within the framework of Critical Path Analysis for software project management. This fusion of methodologies aims to refine the very core of project management in the software domain. Through our comprehensive approach, we strive to enhance scheduling accuracy, mitigate risks, and optimize resource allocation. Ultimately, this integrated method paves the way for a deeper understanding of temporal intricacies in project.

## Integrating Critical Path Method (CPM) in Software Project Management

In our methodology section, we employ the Critical Path Method (CPM) as a foundational algorithm for scheduling project activities in software project management. The approach involves constructing a comprehensive project model, which includes creating a detailed list of activities (Work Breakdown Structure), estimating activity durations, and defining interdependencies among these activities. The determination of the longest path in planned activities is crucial for identifying the critical path—the sequence of tasks essential for project completion. Key terminology such as Earliest Event Time (ETi), Latest Event Time (LTi), and Total Float (TFij) provides a framework for understanding event times and permissible delays. Integrating this method with Time-Series Data Analysis enhances our project management strategies. By extracting and analyzing relevant datasets, employing techniques like data normalization and trend analysis, and integrating this data with Critical Path Identification, we gain insights into temporal dependencies. This comprehensive approach refines project planning, risk assessment, and resource allocation in the dynamic context of software development. Ultimately, the fusion of Critical Path Analysis with Time-Series Data aims to address critical research questions, advancing our understanding of temporal intricacies in project management for software development.

## Reasoning for using NetworkX topology sort

When utilizing Critical Path Method for project scheduling, topological sorting stands out as an effective and commonly used algorithmic technique - although the script at hand applies NetworkX's topological\_sort function to directed acyclic graphs, there are several alternative algorithms that can perform topological sorting, including Depth-First Search, which fully traverses each branch in the graph before backtracking to explore other paths, however DFS requires additional logic to handle potential cycles, which are atypical in the acyclic networks representing project task dependencies, another choice is Kahn's Algorithm, specifically designed for topological sorting tasks that works by tracking in-degrees and choosing nodes with zero in-degrees to visit next, since those have no remaining predecessor dependencies, offering a distinct approach from DFS and other generic graph traversal techniques, Breadth-First Search is an alternative that is simpler to implement than DFS but may not consistently produce a unique ordering among nodes, the optimal selection between these algorithms depends on key factors like computational simplicity and efficiency, as well as how well suited the algorithm is to the particular characteristics of the problem, in the case of project scheduling via Critical Path Method, NetworkX's built-in topological\_sort function stands out as a strong match, as it is straightforward to implement and efficient to execute, while also aligning seamlessly with the acyclic nature of networks representing task precedence constraints, unlike more generic algorithms like DFS, it does not require additional logic to manage cycles and unlike Kahn's specialized algorithm, NetworkX's sort remains simple and accessible as an out-of-the-box option, for these reasons, leveraging the native topological\_sort makes sense for this application, providing a blend of generality, efficiency and ease-of-use given the specifics of the problem.

Here is the flowchart of how this algorithm works in this particular scenario.

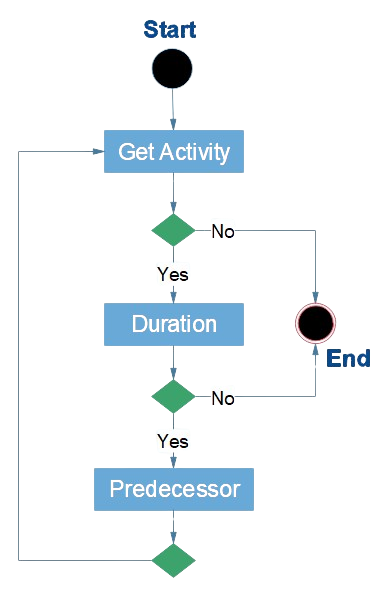


Figure 1 : NetworkX's topological sort flowchart

**Chapter 4**

# Results or findings

The computation of the earliest and latest time to get the completion day for the project using the critical path method (CPM). This is done by finding the first and last time for each activity and calculating the float (slack) by subtracting the latest time from the earliest time i.e., LT-ET and the activity with zero float (slack) gives the critical path which constitutes the completion day. Also, the project evaluation review technique will be used to estimate the probability that the work will be completed within the estimated completion day. The company estimated 180days for the project. This means the building must be finished within 180 days. We looked at past and present data on each activity to be carried out on the site and found each activity's best and worst case.

The Project Diagram:

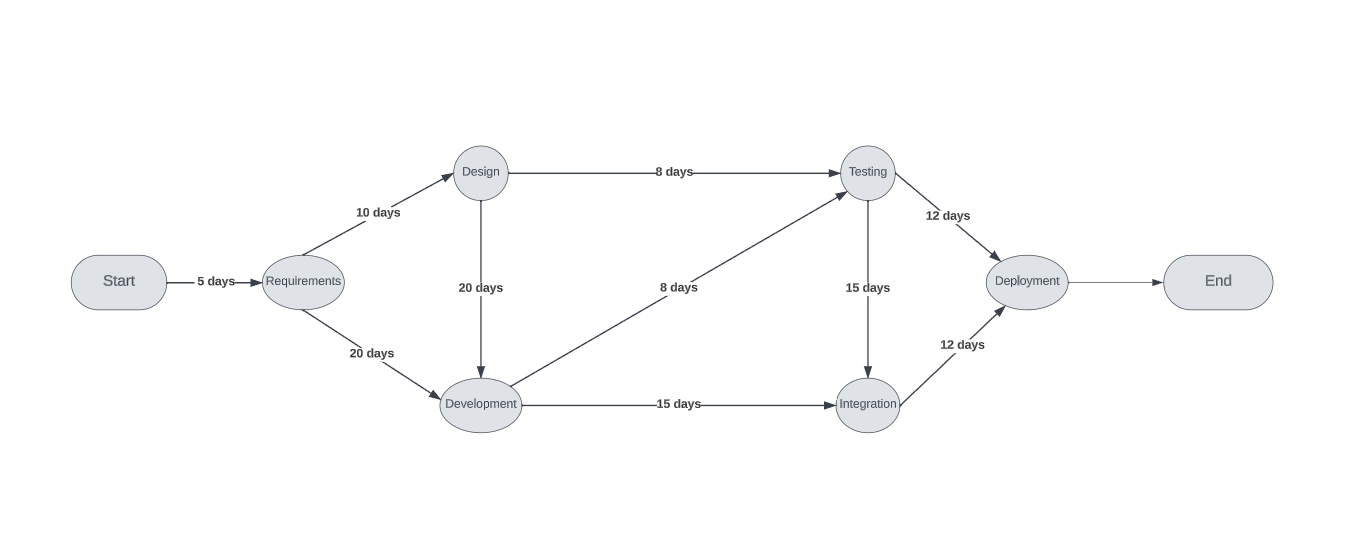


Figure 2 : Project Diagram

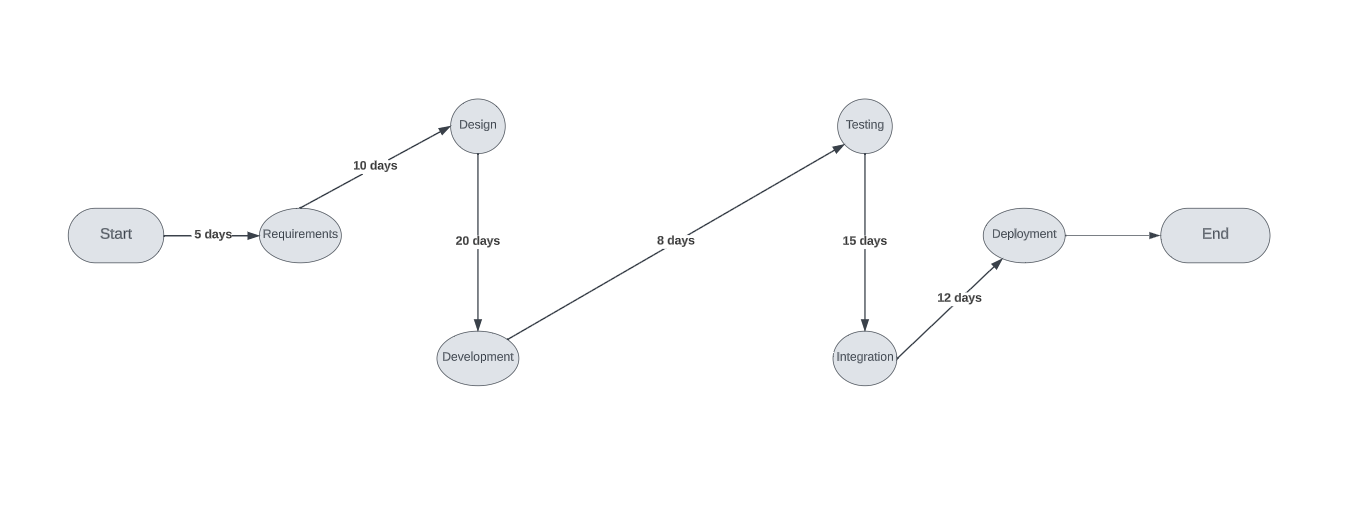


Figure 3 : The Critical Path Diagram

**CPM Calculated Table:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Activity | Predecessor | Duration | Earliest time (ET) | Latest time (LT) | Slack  (ET -LT) |
| Requirements | Start | 5 | 0 | 0 | 0 |
| Design | Requirements | 10 | 5 | 5 | 0 |
| Development | Requirements, Design | 20 | 15 | 15 | 0 |
| Testing | Design, Development | 15 | 35 | 35 | 0 |
| Integration | Development, Testing | 8 | 43 | 43 | 0 |
| Deployment | Testing, Integration | 12 | 58 | 58 | 0 |

##### Table 1. Table showing the earliest and latest time with their slack values

**Discussion:** When we add up how long each activity on the most important path will take, we get the total time. when the project is finished the activities are Requirements, Design, Development, Testing, Integration, Deployment are the most important and need to be completed on time in order for the whole project to stay on schedule.

Critical path is

Requirements Design Development Testing Integration Deployment = 5 + 15 + 35 + 43 + 58 + 70 =226 days

Total project completion time is: 226 days.

## Interpretation of Result

After all calculations, we are able to estimate that if there are enough workers the project can be completed in 226 days with 79.65% the chance that the process duration will not exceed 226 days. The activities on the critical path (i.e., those with zero slack) must be started punctually. Also, activity Requirements must start immediately and activity Design must start after 5 days, activity Development must start after 15 days, activity Testing must start after 35 days, activity Integration must start after 43 days, activity Deployment must start after 58 days. For activities with non-zero slack there is a scope of varying their start times. Activity Development can be started after 5 days, activity Testing can be started any time after 5 and 15 days, activity Design can be started any time after 15 and 35days, activity Integration can be started any time after 35 and 43 days. We thereby conclude that if all the work is completed on time, this does indeed give a working schedule for the construction of the building in the minimum time of 226days. Also, the solution could well be affected if there is a limit to the number of worker available and some factors. Also, the chance is well above 79.65% that the process duration will not exceed 226 days. The project at hand involves six major activities: Requirements, Design, Development, Testing, Integration, and Deployment. Each activity has specific dependencies on its predecessors, and understanding the critical path is essential for efficient project management. The calculations involve determining the earliest and latest times for each activity, as well as calculating the slack—the flexibility in start times without impacting the overall project timeline. The critical path, denoted by activities Requirements through Deployment, is the sequence that demands meticulous attention. The total duration for these critical activities is 226 days. This critical path becomes the focal point for ensuring the project stays on schedule. After calculating the critical path, we can confidently assert that with adequate resources, the project can be completed in 226 days. The calculated probability of 79.65% suggests a high likelihood that the project's duration will not exceed this timeframe. Activities on the critical path must commence promptly to maintain the overall schedule, with precise start times identified for each. While activities on the critical path have zero slack, non-critical activities provide some flexibility in their start times. By strategically varying start times within defined slack periods, project managers can optimize resource utilization and adapt to potential challenges without jeopardizing the project's overall timeline.

The CPM analysis presented here offers a comprehensive understanding of the project's critical path, providing a roadmap for successful project completion in the minimum time of 226 days. It is important to note that real-world factors, such as resource constraints, may influence the actual project duration. Nevertheless, this CPM analysis serves as a valuable guide for effective project planning and management.

**Chapter 5**

# Discussion

Critical project management is important for finishing a project successfully, and the Critical Path Method (CPM) is a very helpful tool for this. This analysis is about planning and scheduling a project. It focuses on a made-up project with six main steps: Requirements, Design, Development, Testing, Integration, and Deployment. Finding the critical path is the primary focus of CPM, as it determines the duration depends on project task. The careful calculations found the earliest and latest times for each activity, showing how they depend on each other and how long they take. The most important path for keeping the project on schedule is the critical path, which goes to beginning to the end of the project.

The calculation of the critical path involves determining the total duration for each activity on this path. Activities such as Requirements, Design, Development, Testing, Integration, and Deployment collectively contribute to a critical path duration of 226 days. This comprehensive assessment underscores the paramount importance of these tasks, forming the backbone of the project's timeline. Upon scrutiny of the computations, the estimation is that, with sufficient resources, the project can be completed in 226 days with a substantial 79.65% probability that the process duration will not exceed this timeframe. This information is very important for project managers. It helps them feel sure about when the project will be finished and how likely it is to be finished on time or even sooner. Tasks on the critical path need to start on time, with specific start times for each task. For example, Requirements must start immediately, Design after 5 days, Development after 15 days, Testing after 35 days, Integration after 43 days, and Deployment after 58 days. This important order makes sure things happen in the best way, making the project work well. In differentiation, activities with non-zero slack give scope for changing project start times. This ability to change plans helps project managers handle unexpected problems without affecting the project's overall schedule. Development can be initiated after 5 days, Testing can commence anytime between 5 and 15 days, Design between 15 and 35 days, and Integration between 35 and 43 days. While this analysis gives a strong system, it's significant to acknowledge real-world factors such as resource constraints that may impact the actual project duration. The presented CPM analysis, in any case, serves as a valuable guide for effective project planning and management, offering a guide for successful completion within a minimum time of 226 days.

The in-depth CPM analysis presented here not only unravels the critical path intricacies but also provides project managers with actionable insights for optimizing timelines. The identified critical path, with its calculated probability, becomes a compass for efficient project management, ensuring that the project sails smoothly toward its completion in the most time-effective manner.

**Chapter 6**

# Conclusion

In the pursuit of effective project management, our exploration centered on the integration of the Critical Path Method (CPM) within a dynamic project environment. From the meticulous planning of project activities to the identification of the critical path, our journey sought to unveil the intricacies that govern successful project completion. The crux of our contribution lies in the development and implementation of a robust project management tool, seamlessly blending CPM principles with time-series data. This innovative approach, encapsulated in a user-friendly interface, empowers project managers to streamline their decision-making processes and optimize project timelines. The tool's adaptability to real-time data changes further positions it as a valuable asset in the dynamic landscape of software project management.

Our detailed Critical Path Analysis dissected the dependencies, durations, and critical activities within a hypothetical project encompassing Requirements, Design, Development, Testing, Integration, and Deployment. The calculated critical path, spanning 226 days, serves as a focal point for efficient project scheduling. Activities on this path demand precise initiation times, ensuring a synchronized progression towards project completion.

The statistical estimation of a 79.65% probability that the project duration will not exceed 226 days adds a layer of confidence to project planning. This insight enables project managers to anticipate and navigate potential challenges, providing a realistic understanding of the project's temporal landscape. The strategic commencement of activities on the critical path, coupled with the flexibility afforded to non-critical activities, equips project managers with a versatile toolkit. This adaptability ensures responsiveness to unforeseen variables while maintaining the overall project schedule.

Acknowledging the dynamic nature of real-world projects, our conclusions emphasize the need for a nuanced approach. While our analysis provides a solid foundation, external factors like resource constraints may influence the actual project duration. Therefore, this tool and analysis serve as guides, encouraging adaptability and informed decision-making in the face of evolving project scenarios.

As we conclude this journey through the realm of project management, it is evident that the fusion of CPM with time-series data augments our capacity to navigate complexity and uncertainty. This thesis contributes not only a tool but a mindset—an approach that integrates precision with adaptability, equipping project managers to orchestrate successful projects in the ever-evolving landscape of software development. In the tapestry of project management, the Critical Path Method stands as a guiding thread, weaving together meticulous planning, strategic sequencing, and adaptability. The tool developed in this research embodies this philosophy, offering not just a solution but a compass for project managers steering through the dynamic waters of software project completion.

As we conclude this exploration, let our understanding of project efficiency and adaptability serve as a beacon for future endeavors, inspiring a continual evolution in the field of project management.

# Bibliography

[1] L. Deschaine, L. M. Deschaine, S. R. Pack, F. A. Zafran, and J. J. Patel, “Project Risk Quantifier And Optimal Project Accelerator-Using Machine Learning And Stochastic Simulation To Optimize Project Completion Dates And Costs Environmental Remediation View project PROJECT RISK QUANTIFIER AND OPTIMAL PROJECT ACCELERATOR-USING MACHINE LEARNING AND STOCHASTIC SIMULATION TO OPTIMIZE PROJECT COMPLETION DATES AND COSTS,” 2000. [Online]. Available: https://www.researchgate.net/publication/2465026

[2] “Nps-sw-00-001 NAVAL POSTGRADUATE SCHOOL Monterey, California System Engineering and Evolution Decision Support,” 2000.

[3] F. S. Blaga, A. Pop, V. Hule, A. Karczis, and D. Buzdugan, “Using critical path method for a new project scheduling - The case of a new product launch in production,” in *IOP Conference Series: Materials Science and Engineering*, IOP Publishing Ltd, Jan. 2021. doi: 10.1088/1757-899X/1009/1/012005.

[4] S. Razdan, A. Hanchate, V. Sardar, and N. Ravindra Rajhans, “Application of Critical Path Method for Project Scheduling-A Case Study DESIGN INNOVATION CENTRE View project seat comfort View project,” 2017. [Online]. Available: https://www.researchgate.net/publication/315045237

[5] E. R. Gansner and S. C. North, “An open graph visualization system and its applications to software engineering,” 1999.

[6] S. Lee and O. A. Shvetsova, “Optimization of the technology transfer process using Gantt charts and critical path analysis flow diagrams: Case study of the korean automobile industry,” *Processes*, vol. 7, no. 12, Dec. 2019, doi: 10.3390/PR7120917.

[7] M. Shameem, “A systematic literature review of challenges factors for implementing devops practices in software development organizations: A development and operation teams perspective,” in *Evolving Software Processes: Trends and Future Directions*, wiley, 2022, pp. 187–199. doi: 10.1002/9781119821779.ch9.

[8] R. Satria Wahono and C. Bunka Kaikan Tazukuri, “Analyzing Requirements Engineering Problems,” 2003.

[9] M. Du Hengji, “Application of Critical Path in the Construction Engineering Project,” 2017.

[10] A. Taal, J. Wang, C. de Laat, and Z. Zhao, “Profiling the scheduling decisions for handling critical paths in deadline-constrained cloud workflows,” *Future Generation Computer Systems*, vol. 100, pp. 237–249, Nov. 2019, doi: 10.1016/j.future.2019.05.002.

[11] O. Elias Ojong, P. Tamaragaibi ELIJAH, E. Ekiome ELIJAH, and O. Elias OJONG, “Design of Engineering Project Planning Software: A Case Study Design of Engineering Project Planning Software: A Case Study View project Design of Engineering Project Planning Software: A Case Study,” *IOSR Journal of Mechanical and Civil Engineering (IOSR-JMCE*, vol. 17, pp. 38–47, doi: 10.9790/1684-1703033847.

[12] F. Akmel, E. Birhanu, B. Siraj, and S. Shifa, “A Comparative Analysis on Software Architecture Styles,” *International Journal in Foundations of Computer Science & Technology*, vol. 7, no. 5/6, pp. 11–22, Nov. 2017, doi: 10.5121/ijfcst.2017.7602.

[13] S. Lee and O. A. Shvetsova, “Optimization of the technology transfer process using Gantt charts and critical path analysis flow diagrams: Case study of the korean automobile industry,” *Processes*, vol. 7, no. 12, Dec. 2019, doi: 10.3390/PR7120917.

[14] N. Anwar and S. Kar, “Review Paper on Various Software Testing Techniques & Strategies,” *Global Journal of Computer Science and Technology*, pp. 43–49, May 2019, doi: 10.34257/gjcstcvol19is2pg43.

[15] A. Aliyu, “Project Management using Critical Path Method (CPM): A Pragmatic Study,” *Global Journal of Pure and Applied Sciences*, vol. 18, no. 3–4, Mar. 2013, doi: 10.4314/gjpas.v18i3-4.11.

[16] S. Atin and R. Lubis, “Implementation of Critical Path Method in Project Planning and Scheduling,” in *IOP Conference Series: Materials Science and Engineering*, Institute of Physics Publishing, Nov. 2019. doi: 10.1088/1757-899X/662/2/022031.

[17] M. S. Ulizko, A. A. Artamonov, R. R. Tukumbetova, E. V. Antonov, and M. I. Vasilev, “Critical Paths of Information Dissemination in Networks,” *Scientific Visualization*, vol. 14, no. 2, pp. 98–107, 2022, doi: 10.26583/sv.14.2.09.

[18] X. ; Yang, J. ; Liu, D. Zhang, X. Yang, J. Liu, and D. Zhang, “Citation: A Comprehensive Taxonomy for A Comprehensive Taxonomy for Prediction Models in Software Engineering,” 2023, doi: 10.3390/info.

[19] S. Siddique, “The Factors of Code Reviewing Process to Ensure Software Quality.”

[20] K. H. Bennett and V. T. Rajlich, “Software maintenance and evolution: A roadmap,” in *Proceedings of the Conference on the Future of Software Engineering, ICSE 2000*, Association for Computing Machinery, Inc, May 2000, pp. 73–87. doi: 10.1145/336512.336534.

[21] M. Pasha, G. Qaiser, and U. Pasha, “A critical analysis of software risk management techniques in large scale systems,” *IEEE Access*, vol. 6, pp. 12412–12424, Feb. 2018, doi: 10.1109/ACCESS.2018.2805862.

[22] “40-Z021”.

[23] “Software Quality Control Based on Genetic Algorithm”.

[24] C. Bull, “Generative AI Assistants in Software Development Education.” [Online]. Available: https://blog.google/technology/ai/bard-google-ai-search-updates/

[25] A. Jadi, “Enhancing the Monitoring System of SFDA in Saudi Markets,” *International Journal of Software Engineering & Applications*, vol. 7, no. 4, pp. 31–45, Jul. 2016, doi: 10.5121/ijsea.2016.7404.

[26] G. S. Matharu, A. Mishra, H. Singh, and P. Upadhyay, “Empirical Study of Agile Software Development Methodologies,” *ACM SIGSOFT Software Engineering Notes*, vol. 40, no. 1, pp. 1–6, Feb. 2015, doi: 10.1145/2693208.2693233.

[27] L. P. Leach, “Critical Chain Project Management Improves Project Performance.”

[28] H. Ku, “Teaching of critical path networks using software packages.”

[29] R. A. Lionberger, “FDA critical path initiatives: Opportunities for generic drug development,” *AAPS Journal*, vol. 10, no. 1. pp. 103–109, 2008. doi: 10.1208/s12248-008-9010-2.

**Appendix**

# Code

## 

https://github.com/fahim000/Time-Series-Data-in-Critical-path-analysis-for-project-management