QUANTITATIVE RISKS ANALYSIS ON SOFTWARE PROJECTS USING THE MONTE CARLO METHOD

КІЛЬКІСНИЙ АНАЛІЗ РИЗИКІВ ПРОГРАМНИХ ПРОЕКТІВ ЗА МЕТОДОМ МОНТЕ-КАРЛО

by Yuriy Oliyarnyk

Presented in Partial Fulfillment of the Requirements for the Degree

Master of Software Engineering

American University Kyiv

2024

APPROVED BY:

Viktor Putrenko, Dr. Habil., Prof.

Acknowledgments

I would like to express my heartfelt gratitude to my scientific advisor, Dr. Viktor Putrenko, for their unwavering support, guidance, and mentorship throughout the course of this research. Dr. Putrenko has been instrumental in shaping the direction of this work, providing invaluable insights, and fostering an environment of intellectual curiosity. Their expertise and dedication have significantly contributed to the depth and quality of this research endeavor.

I am also indebted to Dr. Oleksandra Antoniuk for her keen interest in this project and for posing thought-provoking questions during the preview stages. Dr. Antoniuk's intellectual curiosity and insightful inquiries have played a crucial role in refining the focus of this research and enhancing its overall rigor.

I extend my sincere appreciation to Dr. Jacek Leskow**,** Dean of EPAM School of Digital Technologies for creating an enriching academic environment and encouraging me to explore innovative approaches within the scope of this study.

Thank you for your unwavering support, encouragement, and intellectual guidance.

Yuriy Oliyarnyk, 12/31/2023

Table of Contents

List of Figures

List of Tables

List of Abbreviations

CPM	Critical Path Method
PMI	Project Management Institute
AI	Artificial Intelligence

1. Introduction

1.1. Background

In today's turbulent business environment, software projects are essential for fostering creativity, productivity, and overall organizational performance. However, a number of hazards are introduced by the intrinsic complexity of software development projects, which may affect their effective completion.

Although the utilization of modern project management tools and techniques established a clear trend of improving the success rate of the projects in the recent years [1-3], the landscape of project management has been marked by persistent challenges, as evidenced by substantial investment waste and project failures. In 2020, a disconcerting revelation emerged, indicating that an average of 11.4 percent of investment was squandered due to poor project performance [4]. This alarming trend persisted into 2021, with a staggering 33 percent failure rate observed in IT projects, resulting in significant budget losses [5].

As organizations ventured into 2022, the challenges in project management seemed to intensify, particularly in the realms of big data projects, analytics, and Artificial Intelligence (AI). Disturbingly, the failure rates for projects in these domains loomed large at a staggering 85 percent [6]. These statistics highlight the urgency for a recalibration of project management strategies to address the evolving complexities and uncertainties associated with contemporary technological endeavors.

The years-long trend of less-than-ideal project results highlight the urgent need for a paradigm change in project management techniques. Organizations must prioritize implementing efficient project management techniques in order to reduce the ongoing risk of investment waste and project failures. By using lessons learned from the past, they can develop more comprehensive strategies for the future.

In recent years, there has been an increase in the use of quantitative risk analysis techniques to lower uncertainty in software projects. One such technique that is gaining popularity is the Monte Carlo approach, which is a statistical tool for simulating the probability of different outcomes in a process. This capstone project's goal is to enhance decision-making and project management practices by researching and utilizing the Monte Carlo method for quantitative risk analysis in software projects.

1.2. Objectives of the capstone project

The primary objective of this capstone project is to build a software for conducting a quantitative risk analysis on software projects using the Monte Carlo method. This involves using statistical models to simulate various project scenarios and estimating the likelihood of different outcomes. By using this software, project managers can enhance the accuracy of risk assessment and make more informed decisions and allocate resources more effectively.

This capstone project is significant in addressing the gaps in current risk management practices in software development. By incorporating quantitative analysis through the Monte Carlo method, it seeks to provide a more robust framework for identifying, assessing, and mitigating risks. The findings from this study can contribute to the advancement of project management methodologies, particularly in the context of software development, where uncertainties and complexities are inherent.

More specifically we consider includes using the Monte Carlo method to calculate project risks for software development. The practical goals of the study are to identify important risk factors, model how they affect project outcomes, and give project managers useful information. Although the concepts investigated in this project can be applied broadly to a range of software development situations, projects with well-defined technical tasks and an organized development process will receive particular attention.

The Monte Carlo technique, renowned for its adaptability in managing uncertainty, shall be utilized to approximate the likelihood of various results. In addition to giving project managers a thorough understanding of potential hazards, this method gives them the ability to make informed decisions that will maximize resource allocation and improve project outcomes. Project managers will be more capable of navigating the complexity of software development by making use of the capabilities of the software produced in this project, which will ultimately result in more effective project executions.

Furthermore, this capstone project's importance goes beyond the direct application of risk analysis in software projects. It aims to fill up the gaps that currently exist in traditional risk management techniques in the field of software development. Traditional risk assessment methods frequently lack the level of precision and objectivity that is introduced by using quantitative analysis through the Monte Carlo method. Therefore, the project's goal is to provide insightful ideas and approaches that can raise the bar for project management in general, especially in the dynamic and unpredictable field of software development. The project's results have the potential to advance risk management techniques, making it possible for software projects in the future to handle uncertainty more effectively and resiliently.

1.3. Brief overview of the Monte Carlo method

The Monte Carlo approach is a well-known statistical tool that is commonly used to estimate uncertainty and examine the range of possible outcomes in many different domains, including project management. The Monte Carlo method offers a comprehensive methodology that surpasses classic deterministic methods in the area of software project time estimation.

Fundamentally, the Monte Carlo approach models the inherent uncertainties related to project parameters by creating random samples from probability distributions. These factors could include job completion times, resource availability, and interdependence between various project activities in the context of estimating the duration of software projects. Project managers can use the Monte Carlo approach to simulate a variety of project scenarios and determine the probability of reaching particular project milestones within predetermined timeframes by including these factors into the model.

The first step for project managers using the Monte Carlo approach for software project duration estimation is to define the probability distributions for all relevant parameters. To ascertain the range and probability of possible values for each variable, this may entail the examination of past data, professional judgment, or other information sources. The process then use random sampling techniques to provide many iterations of the project timetable that take various elements into account.

A probability distribution of project durations is the result of the Monte Carlo simulation, which offers a more thorough and nuanced picture of the possible project completion dates. Project managers can determine the range of potential durations and the corresponding probabilities, in addition to the most likely duration, using this distribution. Project managers may proactively manage risks throughout the software development lifecycle, set reasonable expectations, and make better decisions with this information at their disposal.

One notable advantage of the Monte Carlo method is its ability to handle complex project structures and dependencies. As software projects often involve intricate networks of tasks and activities, the method excels in capturing the dynamic nature of these relationships. Additionally, the Monte Carlo method accommodates uncertainties and variability, making it particularly suited for the inherently unpredictable nature of software development.

To sum up, the Monte Carlo method offers a reliable and adaptable way to estimate the length of software projects. This approach allows project managers to better plan the entire project, allocate resources more effectively, and make more informed decisions by accepting uncertainty and randomness. The Monte Carlo approach is a useful tool to manage the uncertainties in project duration estimation as software projects continue to grow in complexity and scale.

2. Literature Review

2.1. Monte Carlo method research

Any project's ability to manage time effectively is essential to its success, and the Monte Carlo approach has proven to be an effective tool for managing schedule uncertainty. The Monte Carlo method, according to Kroese et al. [13], is a computational process that uses repeated random sampling to get numerical results, making it a flexible method that may be used in a variety of contexts. The approach, which has its roots in the work of mathematician Stanislaw Ulam from the Manhattan Project [14], has been used in project management to improve risk analysis, sampling, estimating, and optimization.

The Monte Carlo method is based on the idea of repeated random sampling, as [13] emphasizes, and it offers a methodical way to estimate numerical results. The method's link with the glitz of Monaco's casinos is not coincidental; rather, it is a reflection of the probabilistic nature of the Monte Carlo simulations and the inherent randomness and unpredictability implicit in the outcomes of games such as dice and roulette.

The Monte Carlo simulation is a technique that is defined in project management literature as involving the iteration of project models with random selections from probability distributions for variable input values, such as cost estimates or activity durations [8]. Real-time applications of the Monte Carlo method are frequently neglected, despite the method's potential benefits in resolving scheduling and cost management difficulties [16]. This method's limited adoption is ascribed to project managers' uneasiness with advanced statistical methodologies and their lack of comprehension; they see it more as a challenge than as a means of improving project outcomes.

However, there are useful uses for the Monte Carlo approach in project time management, especially when estimating the risks related to estimated costs and completion schedules [17]. It is a useful tool in project scheduling because of its capacity to deliver insights with a certain level of dependability and to disclose the likelihood of hitting scheduled milestones [19]. Quantitative risk analysis is based on the Monte Carlo simulation of project schedules, which enables project managers to objectively determine schedule reserves and assign probability distribution functions to activity durations [20].

Monte Carlo simulation is a technique that involves many calculations of the project model using random variable input values derived from probability distributions, according to the Project Management Institute (PMI) [8]. Although [16] outlines several problems that impede the use of the Monte Carlo method, it is positioned as a valuable tool for risk management, notably in the areas of time and cost. The method is underutilized and, at times, seen as a complicating element rather than a facilitator due to project managers' reluctance to accept advanced statistical methodologies and a lack of knowledge.

Despite these challenges, the Monte Carlo method finds resonance in project time management, offering a means to quantify risks related to budgets and completion times. Thus, authors in [17] affirm its relevance in providing insights into the likelihood of meeting planned milestones and the associated costs. Williams (2003) emphasizes its role in offering a probabilistic view of project outcomes, allowing project managers to make informed decisions about scheduling and resource allocation.

The use of the Monte Carlo simulation to determine the likelihood of attaining goal completion dates is one way that it is applied in project time management. As noted by Salkeld (2016), Vanhoucke (2016), and Wanner (2013), the quantitative risk analysis methodology is based on the simulation of project schedules utilizing this approach. As part of the process, probability distribution functions are assigned to each project activity. The three-point estimate—optimistic, most likely, and pessimistic—is frequently used to model uncertainty. By using these metrics in the Monte Carlo simulation, a more impartial foundation for scheduling reserves is made possible.

A case study is used to demonstrate how the Monte Carlo simulation is actually used in project time management. Oracle Crystal Ball is used as the simulation program, as seen in this hypothetical project data that was taken from [23]. The program makes it easier to choose inputs from presumptive distributions for every task, producing a result variable frequency distribution. This result helps with decision-making by offering insightful information about the likelihood of meeting deadlines or completing the project earlier than anticipated.

In conclusion, the Monte Carlo method's application in project time management offers a systematic and probabilistic approach to address uncertainties. While challenges persist in terms of understanding and adoption, the method's potential to enhance risk analysis, provide insights into project outcomes, and facilitate informed decision-making positions it as an asset in the project management toolkit.

Interesting literature review of Monte Carlo methos is conducted in [12] which was our guideline in our work.

2.2. Known software implementations

In the dynamic landscape of project management, where uncertainties and risks are inherent, the Monte Carlo method stands out as a powerful tool. Numerous software implementations of the Monte Carlo method have emerged, offering diverse functionalities and features to address the complex challenges associated with project time management. From widely used platforms such as Oracle Crystal Ball and Primavera Project Planner to specialized tools like @RISK, Tamara, Safran Risk, and ModelRisk, the spectrum of available software reflects the diversity of approaches in handling probabilistic elements within project scenarios. This introduction sets the stage to explore the myriad software implementations of the Monte Carlo method, emphasizing their collective contribution to enhancing risk assessments, optimizing resource allocation, and fostering resilience in the ever-evolving landscape of project management.

2.2.1 Oracle Crystal Ball

Oracle Crystal Ball [24] is widely used for project time management using the Monte Carlo method. It offers a user-friendly interface, allowing project managers to define probability distributions, conduct simulations, and analyze outcomes without needing extensive statistical expertise. However, its cost may be prohibitive for smaller projects, and users might face a learning curve, given its extensive capabilities.

2.2.2 Microsoft Project

To use Monte Carlo simulations with Microsoft Project, an add-on tool is required [25]. These tools allow users to assign different statistical distributions, perform Monte Carlo simulation, and output results in different formats. They also offer features such as probabilistic or conditional branching.

Dependence on additional plugins for integrating Monte Carlo simulations with Microsoft Project introduces a layer of complexity and potential discomfort for users, despite the widespread popularity of Microsoft Project in project management workflows. While these add-on tools enhance the functionality of Microsoft Project by enabling users to assign various statistical distributions and conduct Monte Carlo simulations, the need for an external plugin introduces an extra step in the process. Users may find this dependency cumbersome as it requires additional installation, configuration, and maintenance, potentially disrupting the seamless flow of their project management tasks.

2.2.3 Primavera Risk Analysis

Oracle Corporation created Primavera P6, a high-performance project management tool. It is well known for its sophisticated project management skills and is a preferred option for businesses in a variety of sectors, including manufacturing, engineering, construction, and oil & gas.

A stand-alone program, Primavera Risk Analysis [26] (formerly called Pertmaster) is a component of the larger Oracle Primavera suite, which also includes Primavera P6. Specialized software called Primavera Risk Analysis is made for project managers who need to analyze and model risks quantitatively. The application of Monte Carlo Simulation capabilities is also included in this.

Although the program is quite functional and allows users to model uncertainty, users who only need simple simulations may find it difficult to use due to its complexity, and smaller businesses may find the cost of license prohibitive.

2.2.4 @RISK

@RISK, a popular add-in for Microsoft Excel [27], facilitates Monte Carlo simulations, allowing project managers to incorporate uncertainty into spreadsheet models. While efficient for Excel-based simulations, it may lack the sophistication of standalone project management software, especially when dealing with large datasets or complex project structures.

2.2.5 Tamara

Tamara is another specialized software program made for Monte Carlo simulations and project risk analysis [28]. It offers tools for assessing results, creating insights, and modeling uncertainty in order to improve project time management. In contrast to more generic project management software, it might have a higher learning curve.

2.2.6 Safran Risk

Safran Risk is a project risk management company that uses Monte Carlo simulation [29]. It lets users evaluate how uncertainty affects project budgets, schedules, and resource availability. Although the program has many useful capabilities, smaller projects with tighter budgets may find it prohibitively expensive. It may also be difficult to integrate the software with current workflows, requiring more time and resources to do so smoothly and integrating the program into the current project management system. These disadvantages should be carefully taken into account, particularly for businesses with tight budgets and specialized workflow designs. It is imperative to evaluate the software's applicability by balancing its advantages over any potential drawbacks and making sure that the expenditure is commensurate with the scope of the project and the flexibility of the current operating procedures.

2.2.7 ModelRisk

ModelRisk is an Excel add-in that focuses on Monte Carlo simulations and risk analysis [30]. Since 2009, it has dominated the industry by providing features that simplify the creation, auditing, and testing of risk models. It replaces uncertain values in Excel models with quantitative probability distribution functions, providing hundreds of possible scenarios. Insights on budget likelihood, investment returns, energy price exposure, capital requirements, and venture success determinants are offered by the results, which are presented in graphical and statistical formats. Because of its smooth integration with Excel, users accustomed to project management using spreadsheets can utilize it. But when compared to stand-alone project management software, its capabilities could be limited, especially when it comes to scalability and collaborative features.

2.3 Conclusions

Current software implementations of the Mote Carlo method exhibit general flaws, which motivates the search for novel alternatives. Cost is still a major obstacle because existing platforms' hefty licensing costs make them inaccessible to smaller businesses or organizations with tighter budgets. Another general problem is that some tools are overly complicated, which discourages users from looking for simpler Monte Carlo simulations with less complicated features. Because of its complexity, there is a need for fresh, intuitive software that is designed for projects with simple criteria. This would minimize the learning curve and provide a more intuitive experience.

Moreover, scalability and collaboration features are generally limited for applications that function as Excel add-ins; this is especially noticeable when managing large datasets or complex project structures.

These general limitations might be overcome by developing stand-alone apps with increased scalability, offering a more adaptable framework for a range of project scenarios. Furthermore, in contrast to more all-encompassing project management software, the presence of specialist tools could result in a more challenging learning curve due to their narrowly targeted value. This underscores the need for innovative solutions that blend specialized functionality with ease of use, making Monte Carlo simulations accessible to a broader variety of project managers.

In summary, there is a need to investigate other approaches due of the general shortcomings found in the software implementations of the Monte Carlo method, which include complexity, cost issues, and scalability constraints. These proposed solutions could tackle these general problems by providing affordable, approachable, and adaptable platforms fit for a wide range of project management requirements.

3. Problem Statement

3.1. Project Duration Estimation

In the current economic landscape, characterized by rapid technological advancements and global uncertainties, the accurate estimation of IT project durations holds unparalleled significance. Precise project timelines serve as a cornerstone for effective resource allocation, budget planning, and strategic decision-making. In times of economic turbulence and crises, the ability to reliably predict project durations becomes even more critical, providing organizations with the foresight needed to navigate challenges efficiently. Proper estimation not only aids in mitigating potential risks but also enhances project resilience, ensuring that IT initiatives can adapt to unforeseen circumstances. In this context, the prudent management of IT project timelines acts as a strategic imperative, allowing businesses to optimize resources and remain agile in the face of economic uncertainties.

Our core goal in this capstone project is to refine and improve *project duration estimation* by synergizing the Critical Path Method (CPM) with Monte Carlo Simulation. The central dilemma revolves around the challenge of accurately predicting project timelines, particularly in scenarios where activities are fraught with inherent risks. The project aims to deploy the CPM, a well-established scheduling methodology, to construct a robust model that encompasses the dependencies and durations of interconnected activities.

A pivotal aspect of this project involves the infusion of Monte Carlo Simulation, introducing a probabilistic dimension into the realm of project duration estimation. This entails leveraging optimistic, most likely, and pessimistic values for activity durations, all rooted in an assumed risk function. The fundamental premise guiding this approach is the recognition that activity durations inherently carry uncertainties, displaying variability within predefined ranges due to unforeseen circumstances.

The practical significance of this endeavor lies in its response to the limitations of deterministic methods, particularly the traditional CPM, which may fall short in capturing the dynamic nature of real-world project environments. By factoring in the inherent risks associated with activity durations, the project aspires to present a more nuanced and realistic paradigm for project management. This, in turn, equips project managers with enhanced tools for decision-making in the face of uncertainties.

In summary, this capstone project is a concerted effort to address the pressing challenges faced by project managers in navigating uncertainties during project scheduling. The amalgamation of CPM and Monte Carlo Simulation is anticipated to yield a comprehensive, adaptable solution, better suited to handle the intricacies and uncertainties inherent in the process of project duration estimation.

3.2. Critical Path Method

The Critical Path Method (CPM) stands as a pivotal technique in project management, particularly in the realm of project duration estimation [31,32]. Operating on the principle of identifying the longest path through a network of dependent activities, CPM plays a crucial role in determining the minimum time required for project completion. At its core, CPM involves a meticulous analysis of all project activities, considering their dependencies and durations, to establish a comprehensive project schedule. The identification of the critical path, which comprises activities with zero slack or float, denotes the sequence of activities that, if delayed, would extend the project duration.

The estimation of project duration using CPM follows a deterministic approach, relying on the assumption of fixed activity durations. Each activity is assessed in terms of its earliest start and finish times, latest start and finish times, and total slack or float. The critical path is then derived by connecting activities with zero slack, representing the sequential path that determines the minimum time required for project completion.

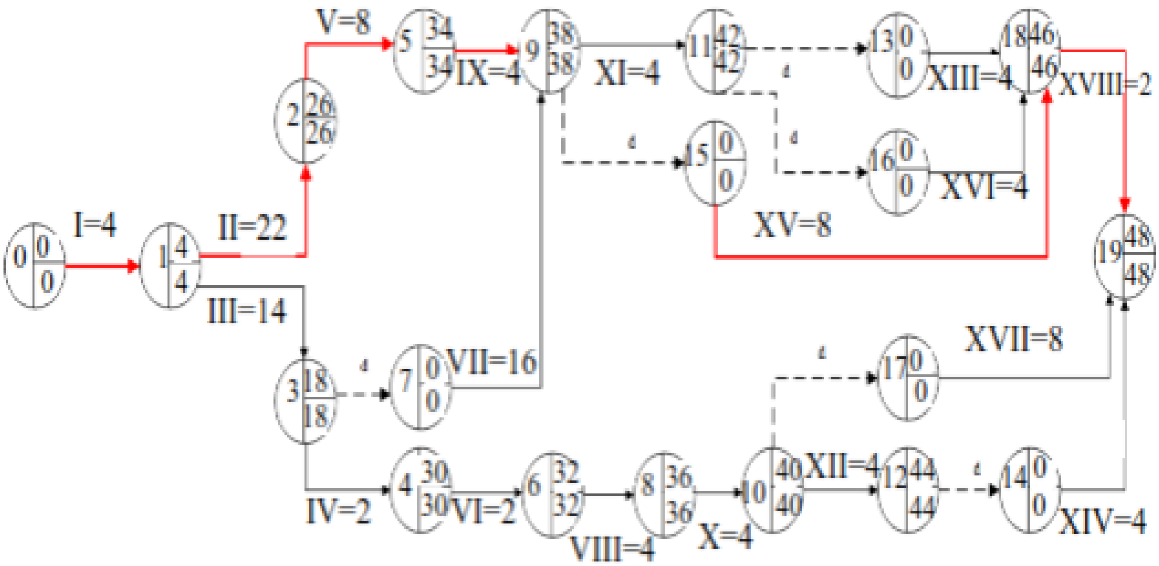


Fig. 1. Example Project Network Diagram. Taken from [31]

While CPM provides a valuable deterministic baseline for project duration, it inherently assumes that activity durations are fixed and known with certainty. In real-world scenarios, uncertainties and risks surround project activities, making it imperative to enhance the accuracy of duration estimates. This is where the integration of probabilistic methods, such as Monte Carlo Simulation, becomes crucial.

Critical Path Method serves as a fundamental tool for project managers to delineate the most critical sequence of activities, establishing a baseline for project duration estimation [7-9]. However, its deterministic nature calls for a complementary approach, such as Monte Carlo Simulation, to account for uncertainties and enhance the robustness of project timelines in dynamic project environments. The synergy of CPM and probabilistic methods equips project managers with a comprehensive toolkit to navigate uncertainties and ensure successful project delivery.

4. Monte Carlo Method Implementation

4.1. Solution architecture.

The C4 architecture views [33] provide a comprehensive framework for visualizing and documenting the architecture of complex software systems. These views consist of multiple levels, each offering a distinct perspective on the system's structure and behavior. At the top level, the Context Diagram (Level 1) provides an overarching view, highlighting the system's boundaries, external actors, and the high-level interactions between them. The Container Diagram (Level 2) dives deeper, focusing on software containers such as web applications, databases, and microservices, along with their interactions. The Component Diagram (Level 3) delves into the internal components of containers, showcasing classes, interfaces, and their relationships. Finally, the Code Diagram (Level 4) allows for code-level insights, revealing individual methods, functions, and their connections. The C4 architecture views offer a systematic approach to understanding and communicating software architecture, making them invaluable tools for architects, developers, and stakeholders.

4.1.1. Component Diagram

The Component Diagram at Level 3 of the C4 architectural model provides a detailed view of the software system, focusing on the individual components that make up the system. This diagram highlights the internal structure of each container, breaking it down into its constituent components, such as microservices, modules, classes, or functions. Each component is represented as a box within the container it belongs to, and connections between components illustrate the dependencies and interactions between them. For the purpose of architecture definition we prefer to use the Component Diagram at Level 3 because it allows us to gain a comprehensive understanding of the internal workings of the system. It provides valuable insights for developers, architects, and stakeholders, aiding in the identification of bottlenecks, potential performance issues, and areas for optimization. With this detailed view, teams can make informed decisions about system design, scalability, and maintainability, ensuring that the software system aligns with architectural goals and effectively delivers its intended functionality.

You can find the Component Diagram of the system on Fig. 2.

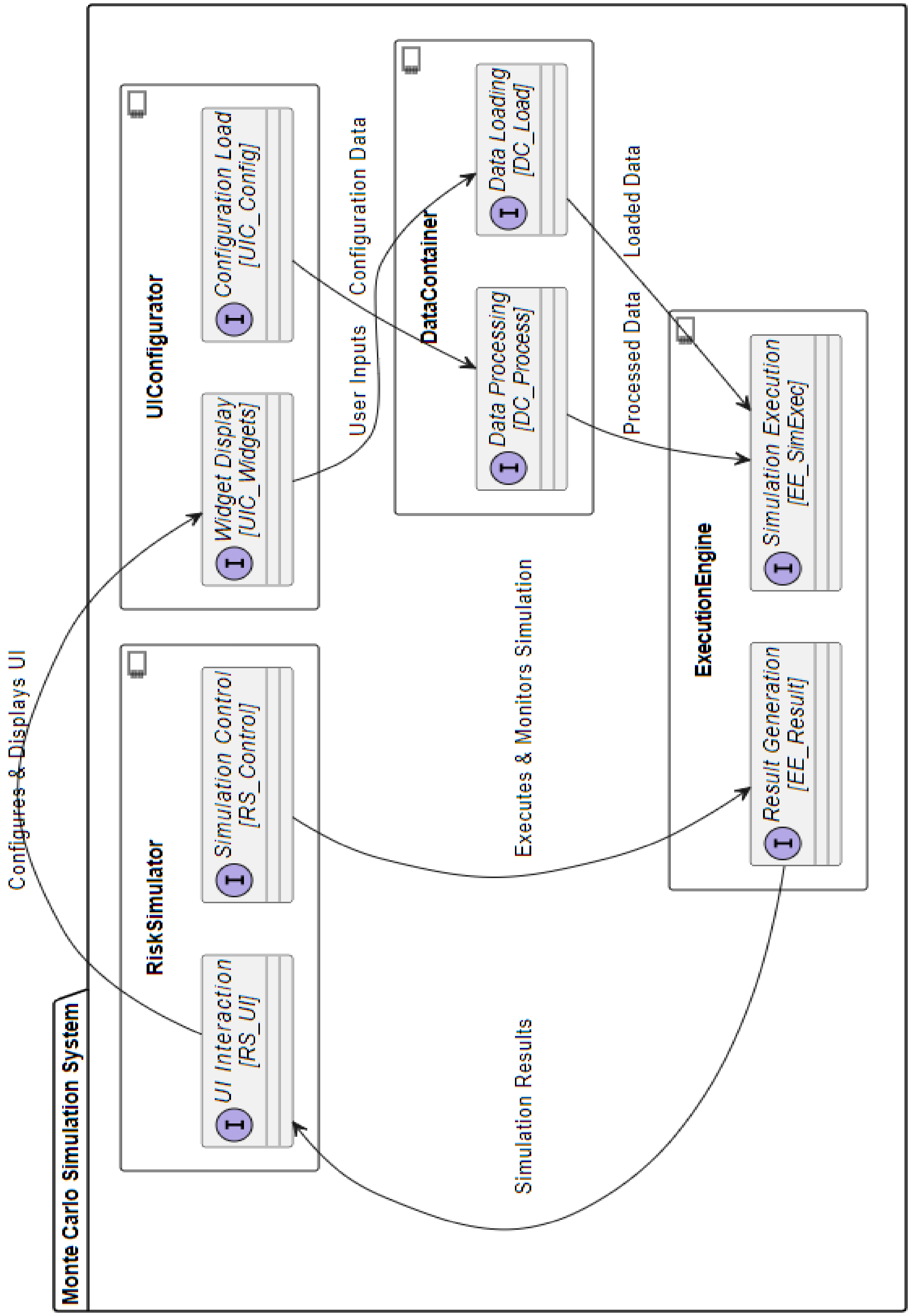


Fig.2. Architecture: Component Diagram

4.2. Technology stack

Following technologies were used for creating software:

Table 1. Technology stack for Monte Carlo implementation

Parameter	Value
Programming Language	Python v.3.8
Development Environment	Jupiter Notebooks
Input Data Format	Microsoft Excel

The strategic choice of Python and Jupyter Notebook as the technology stack for the capstone project not only reflects contemporary industry preferences but also offers a range of compelling benefits. Python's popularity in the data science and machine learning domains stems from its extensive libraries such as NumPy, Pandas, and Scikit-learn, providing robust tools for data manipulation, analysis, and modeling. The language's clear syntax and readability contribute to a more efficient and collaborative coding environment. Jupyter Notebook, with its interactive and iterative nature, amplifies the advantages of Python by offering a seamless integration of code, visualizations, and narrative documentation. This enables a transparent and accessible project development process, where stakeholders can actively engage with the project's methodology and findings. The modular structure of Jupyter Notebook supports incremental development, making it conducive to quick testing and validation of hypotheses. In essence, the Python and Jupyter Notebook stack not only aligns with contemporary technological trends but also facilitates a dynamic, collaborative, and efficient implementation of the capstone project.

4.3. Input Data Specification

4.3.1. Activities Dataset

Our solution expects Project Activities data are provided in the Excel format of appropriate format described below. There are two different estimate types are supported:

- Single point estimate (just one number representing effort in hours)
- Three-points estimate (three numbers representing efforts in hours for 3 cases : optimistic, most-likely, and pessimistic)

Tables 2 and 3 below provide a concise description of the columns in the Excel file used as input for the Monte Carlo implementation in Project Time Duration Calculation. Each column serves a specific purpose in capturing essential details about project activities, including effort estimates, unique identifiers, names, complexity levels, predecessors, assignments, and estimations. This structured format facilitates a clear understanding of the input data, ensuring seamless integration into the Monte Carlo simulation process.

Besides, sample data file named *project-activities.xlsx* provided as a part of the solution contains both cases on different tabs, named *'Single-Point Estimations'* and *'Three-Point Estimations'* correspondingly.

Screenshots of sample data are provided below on Fig.2 and 3.

American University Kyiv
Powered by Arizona State University®

Capstone Project: Sample Data

Activity #	Activity Name	Complexity	Predcessors	Assignment	EFFORT ESTIMATE, man*day Estimation
1	Define Project Scope	Low		John	30.0
2	Research Data Lake Architecture	Medium	1	Alice	55.0
3	Procure Hardware and Software	Complex	2	Bob	75.0
4	Install Data Lake Infrastructure	Medium	3	Charlie	45.0
5	Configure Storage and Security	Low	4	David	20.0
6	Ingest Data Sources	Medium	2,4	Eve	60.0
7	Implement Data Governance	Complex	5,6	Frank	90.0
8	Develop ETL Processes	Medium	6,7	George	55.0
9	Create Data Catalog	Low	8	Helen	30.0
10	Implement Data Quality Checks	Medium	8,9	Ian	65.0
11	Perform System Integration Tests	Complex	10	Jack	85.0
12	Develop User Training Materials	Low	9,11	Kelly	25.0
13	Conduct User Training	Medium	12	Lisa	50.0
14	Implement Monitoring Solutions	Medium	10,13	Mark	60.0
15	Prepare for Go-Live	Low	14	Nancy	35.0
16	Data Lake Go-Live	Complex	15	Oscar	95.0
17	Post-Implementation Review	Medium	16	Peter	50.0
18	Documentation and Knowledge Transfer	Low	17	Quinn	40.0
19	Closeout Project	Low	18	Rachel	30.0
20	Stakeholder Communication	Medium	11,14,17	Sam	70.0
21	Project Reporting	Low	19,20	Tom	40.0
	TOTAL:				**1105**

Fig. 3. Project Activities Dataset, Single Point Estimate case

Table 2. Project Activities Dataset column description – Single Point Estimate case

Column	Description
Activity #	Unique identifier for each project activity.
Activity Name	Descriptive name of the project activity.
Complexity	The complexity level of the activity (Low, Medium, Complex).
Predcessors	Identifies activities that need to be completed before the current one can start.
Assignment	The person or team responsible for the activity.
Estimation	Man-days required for the activity's completion.

Capstone Project: Sample Data

Activity #	Activity Name	Complexity	Predcessors	Assignment	EFFORT ESTIMATE, man*day		
					Estimation	Optimistic	Pesimistic
1	Define Project Scope	Low		John	30.0	24.0	45.0
2	Research Data Lake Architecture	Medium	1	Alice	55.0	40.0	82.5
3	Procure Hardware and Software	Complex	2	Bob	75.0	52.5	112.5
4	Install Data Lake Infrastructure	Medium	3	Charlie	45.0	31.5	67.5
5	Configure Storage and Security	Low	4	David	20.0	14.0	30.0
6	Ingest Data Sources	Medium	5	Eve	60.0	42.0	90.0
7	Implement Data Governance	Complex	1	Frank	90.0	63.0	135.0
8	Develop ETL Processes	Medium	7	George	55.0	38.5	82.5
9	Create Data Catalog	Low	8	Helen	30.0	21.0	45.0
10	Implement Data Quality Checks	Medium	9	Ian	65.0	45.5	97.5
11	Perform System Integration Tests	Complex	10	Jack	85.0	63.0	127.5
12	Develop User Training Materials	Low	6,11	Kelly	25.0	17.5	37.5
13	Conduct User Training	Medium	12	Lisa	50.0	35.0	75.0
14	Implement Monitoring Solutions	Medium	13	Mark	60.0	42.0	90.0
15	Prepare for Go-Live	Low	14,1	Nancy	35.0	24.5	52.5
16	Data Lake Go-Live	Complex	15	Oscar	95.0	66.5	142.5
17	Post-Implementation Review	Medium	16	Peter	50.0	35.0	75.0
18	Documentation and Knowledge Transfer	Low	17	Quinn	40.0	21.0	45.0
19	Closeout Project	Low	18	Rachel	30.0	21.0	45.0
20	Stakeholder Communication	Medium	17,19	Sam	70.0	49.0	105.0
21	Project Reporting	Low	20,19	Tom	40.0	28.0	60.0
	TOTAL:				1105.0	774.5	1642.5

Fig.4. Project Activities Dataset, Three Points Estimate case.

Table 3. Project Activities Dataset column description – Three Points Estimate Case

Column	Description
Activity #	Unique identifier for each project activity.
Activity Name	Descriptive name of the project activity.
Complexity	The complexity level of the activity (Low, Medium, Complex).
Predecessors	Identifies activities that need to be completed before the current one can start.
Assignment	The person or team responsible for the activity.
Estimation	Man-days required for the activity's completion, Most-likely case
Optimistic	Man-days required for the activity's completion, Optimistic case
Pessimistic	Man-days required for the activity's completion, Pessimistic case

4.3.2. Project Risks Dataset

Besides the risks generated by the project activities with uncertainties, our solution also supports project risks which affect the project as a whole. Such risks are applied as a fixed percent of estimate calculated earlier using CPM or other duration calculation method. We are calling such risks as "general risks". Presence of general risk obviously increases the project duration. Each individual general risk R_j is defined by its impact value I_j and the probability of its occurrence p_j:

$$R_j = p_j \cdot I_j \,,$$

there $j=1, \ldots, M$ and M – number of general risks on the project. Please note, impact values I_j are percentages or duration increase, thus $0 \le I_j \le 1$.

In this case we have:

$$E_k = A_k \cdot \sum_{j=1}^{M} (1 + R_j) = A_k \cdot \sum_{j=1}^{M} (1 + p_j I_j)$$

there E_k, $k=1, \ldots, K$ is the duration estimate obtained on the k-th iteration of Monte Carlo simulation, A_k – duration estimate calculated prior to general risks factor application basing on project activities, usually using CPM or other method.

In order to enter risks values into the solution another input file in excel format is used : *project-risks.xlsx* This file contains several tabs and data as shown on the following figure 3 below:

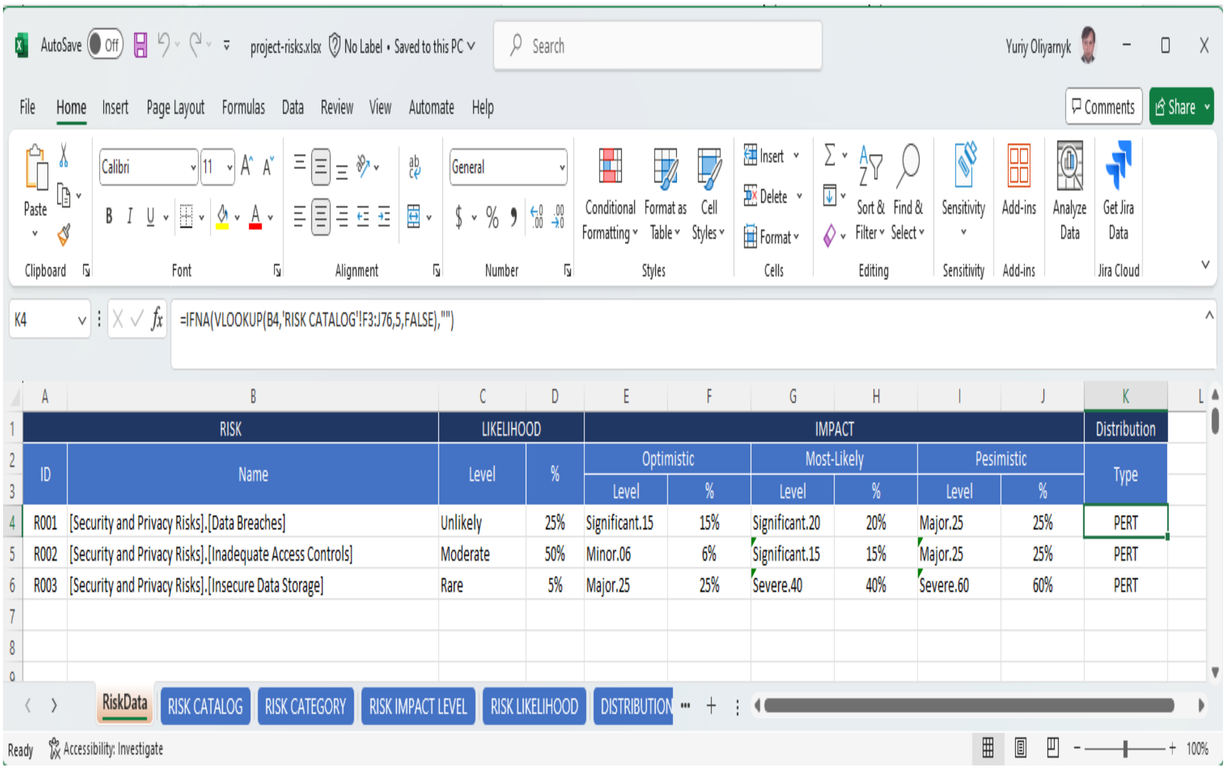


Fig.5. Project General Risks Dataset

Table below contains detailed description for every tab present in the risk definition file.

Table 4. General Risk Definition tabs

Column	Description
Risk Data	Main tab there we define list of individual general project risk

RISK CATALOG	Catalog of risks and their impact percentages. This tab accumulates expert knowledge for a given type of project or business or technical domains.
RISK CATEGORY	Dictionary of risk categories
RISK IMPACT LEVEL	Dictionary of risk impact levels
RISK LIKELIHOOD	Dictionary of risk likelihoods
DISTRIBUTIONS	Dictionary of three-point distributions implemented for risks

Columns descriptions for general risk dataset could be found in the following table.

Table 5. General Risk Dataset Columns Descriptions

Parameter			Description
Risk		ID	Unique identifier of general risk
Risk		Name	Name of given general risk. Chosen from the drop-down list defined by [RISK CATALOG] tab
Likelihood		Level	Risk likelihood level, could be chosen from drop-down list defined by [RISK LIKELIHOOD] tab
Likelihood		%	Auto-calculated % value correspondent to the given likelihood level. Excel formula used for automation
Impact	Optimistic	Level	Auto-calculated level value for optimistic impact level. Excel lookup formula is used to pull appropriate value from [RISK CATALOG] tab
Impact	Optimistic	%	Auto-calculated % value correspondent to the given optimistic impact level. Excel formula used for automation
Impact	Most-Likely	Level	Auto-calculated level value for most-likely impact level. Excel lookup formula is used to pull appropriate value from [RISK CATALOG] tab
Impact	Most-Likely	%	Auto-calculated % value correspondent to the given most-likely impact level. Excel formula used for automation
Impact	Pessimistic	Level	Auto-calculated level value for pessimistic impact level. Excel lookup formula is used to pull appropriate value from [RISK CATALOG] tab
Impact	Pessimistic	%	Auto-calculated % value correspondent to the given likelihood level. Excel formula used for automation
Distribution		Type	Random distribution of general risk event. In this version we have only ‘PERT’ distribution implemented

4.4. Classes

4.4.1. Class UI Configurator

The UIConfigurator class serves as a user interface for configuring parameters essential for conducting a Monte Carlo simulation within a Jupyter notebook environment. Its purpose is to streamline the setup and execution of simulations by providing an organized and interactive interface for users to define various project-related details, including customer and project information, project activities, risk parameters, and simulation settings.

This class encapsulates the logic for creating an interactive and organized user interface, making it user-friendly for configuring parameters essential for Monte Carlo simulations. The dynamic control of widget visibility ensures an intuitive experience, adapting to the specific simulation setup requirements.

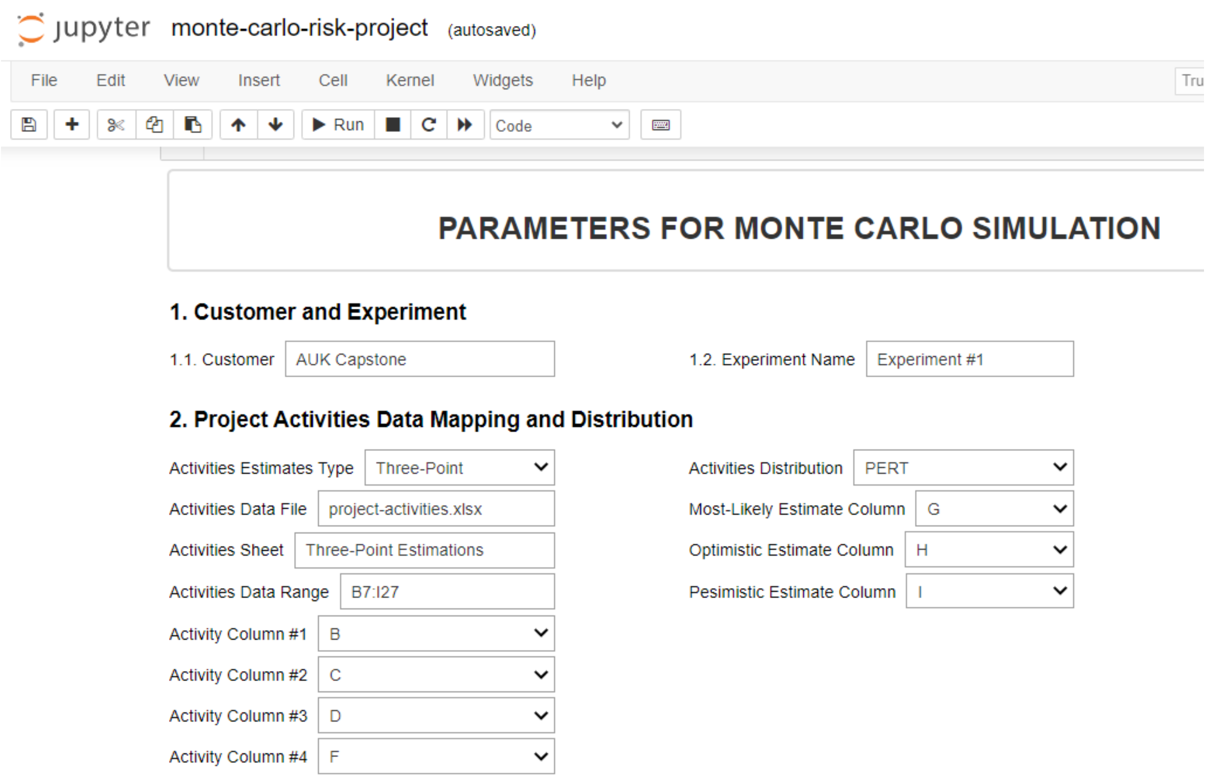


Fig. 6. User Interface represented by UIConfigurator class.

Below is the list of attributes and methods defined in the class.

Attributes:

m_widget_style (dict): A dictionary containing style configurations for widgets.

m_column_list (list): A list of uppercase alphabet letters used for column selection.

config_data (dict): Configuration data loaded from a YAML file, serving as default values for widgets.

m_distributions (list): Default distributions for the simulation.

m_estimate_types (list): Default estimate types for the simulation.

m_percentiles (list): Default percentiles for risk scenarios.

Methods:

__init__(self): Initializes class attributes, loads configuration data from a YAML file, and creates widgets with default values.

handle_estimate_type_change(self, change): Dynamically adjusts the visibility of certain widgets based on changes in the estimate type.

handle_risks_change(self, change): Controls the visibility of widgets related to risk parameters based on changes in the "Include Global Project Risks" checkbox.

display_configuration_form(self): Generates a structured HTML layout for displaying the user interface components, organized into different groups and columns.

The class includes a variety of widgets such as Text, Dropdown, Checkbox, and IntSlider for configuring parameters related to the Monte Carlo simulation. These widgets cover customer and project details, project activities, risk parameters, the number of iterations, and risk scenarios. Certain widgets' visibility is dynamically controlled based on the values of other widgets. For instance, the visibility of risk-related widgets depends on the state of the "Include Global Project Risks" checkbox. Notable that PyYAML is utilized to load configuration data from a YAML file, and the loaded data is used to set default values for the widgets.

4.4.2. Class DataContainer

The DataContainer class is designed to handle and organize project-related data, specifically project activities and risks, loaded from an Excel file. The DataContainer class acts as a container for storing and managing project data, focusing on project activities and risks. It extracts necessary information from the UIConfigurator class and facilitates the loading and validation of project data from Excel files.

Attributes:

- m_project_activities_data (DataFrame): Dataframe to store project activities.
- m_project_risks_data (DataFrame): Dataframe to store project risks.

Methods:

- __init__(self, uiform): Initializes the DataContainer object by extracting relevant information from the UIConfigurator object passed as a parameter. Attributes are populated based on the values of widgets in the user interface.
- simulation_params2pandas(self): Converts simulation parameters to a pandas DataFrame for structured display. Filters out unnecessary parameters related to project data.
- print(self): Prints the simulation parameters in a structured format using Markdown for better visualization.
- is_numeric(self, value): Checks if a given value can be converted to a numeric type.
- __check_row_estimations(self, optimistic, mostlikely, pesimistic): Checks if the estimates provided for a row of project activities are valid.
- load_project_activities_from_xlsx(self): Loads project activities from an Excel file specified in the user interface. Validates and processes the data, storing it in m_project_activities_data.
- __check_row_risks(self, likelhd_pct, imp_opt_pct, imp_mostlikely_pct, imp_pes_pct): Checks if the parameters related to a row of project risks are valid.
- load_project_risks_from_xlsx(self): Loads project risks from an Excel file specified in the user interface. Validates and processes the data, storing it in m_project_risks_data. This method is conditional on the m_risks attribute.

The class uses Pandas DataFrames (m_project_activities_data and m_project_risks_data) to organize and store project-related information. Also class performs validation checks on the input data, ensuring that estimates and risk parameters meet specified criteria.

Data loading methods (load_project_activities_from_xlsx and load_project_risks_from_xlsx) are designed to handle the loading and processing of project data from Excel files.

The class includes error handling mechanisms, printing informative messages when issues arise during data loading.

The simulation_params2pandas method facilitates the display of relevant simulation parameters, filtering out unnecessary attributes related to project data. The print method utilizes Markdown for displaying simulation parameters in a structured format for improved readability.

Overall, the DataContainer class plays a crucial role in managing and preparing project data for subsequent use in Monte Carlo simulations. It ensures data integrity and provides mechanisms for visualizing and validating simulation parameters.

4.4.3. Class ExecutionEngine

The ExecutionEngine class represents the core execution engine for the Monte Carlo simulation of project durations, incorporating uncertainties and risks. This class is responsible for generating simulated project duration values, performing Monte Carlo simulations, and presenting the results.

Attributes:

- target_variable: Represents the target variable for the simulation, set to 'Project Duration'.
- current_run: Represents the current run of the simulation.
- mcrdata: An instance of the DataContainer class containing simulation parameters and input data.
- update_progress_flag: A boolean flag indicating whether to update the progress during the simulation.
- progress_widget: Widget for displaying the simulation progress.

Methods:

- __init__(self, current_run, mcrdata, update_progress_flag, progress_widget): Initializes the ExecutionEngine object with necessary parameters for simulation execution.
- generate_triangular_random(low, high, peak, samples=1): Generates a random variable following a triangular distribution.
- generate_pert_random(low, high, peak, samples): Generates random samples from a PERT distribution.
- generate_random_variable(distribution, optimistic, mostlikely, pesimistic): Generates a random variable based on the specified distribution and estimation values.

- draw_project_cpm_graph(act_df): Draws a Critical Path Method (CPM) graph representing the project's network diagram.

- calculate_simulated_project_duration(new_row): Calculates the simulated project duration by incorporating uncertainties and risks.
- run_monte_carlo_simulation(): Executes the Monte Carlo simulation for project durations.
- output_calculations_data(): Outputs the simulation results, including calculated project durations, to an Excel file.
- display_simulation_results(): Displays simulation results, including project activities, risks, graphs, and summary statistics.
- __plot_tornado_diargam(): Generates and displays a tornado diagram representing the correlation of risk variables with 'value_at_risk'.
- __get_nearest_percentile_value(percentile_value): Private method to retrieve the nearest percentile value from the 'value_at_risk' column of a DataFrame.

The ExecutionEngine class is a pivotal component within the project management simulation framework, orchestrating Monte Carlo simulations to assess project durations. Primarily designed to operate in tandem with the DataContainer class, it encapsulates essential functionalities for generating random variables, simulating project durations under specified uncertainties and risks, and visualizing the results. By leveraging network analysis tools, the class calculates critical paths, offering insights into potential bottlenecks and project completion timelines. Notably, the ExecutionEngine facilitates the exportation of simulation data and provides comprehensive displays of outcomes, including tornado diagrams and percentile tables, enabling project managers to make informed decisions by identifying and mitigating risks associated with project duration uncertainties. In essence, the ExecutionEngine plays a central role in conducting sophisticated Monte Carlo simulations, making it a crucial asset for robust project risk analysis and decision-making.

4.4.4. Class RiskSimulator

The RiskSimulator class serves as the user entry point for facilitating Monte Carlo simulations. This class allows users to, initiate the simulation process, and view the results. The constructor method initializes various UI components by means of creating necessary supportive classes instances.

The 'Run' button triggers the Monte Carlo simulation process, connecting to the 'run_button_click_event_handler' method. Similarly, the 'Exit' and 'Clear' buttons are linked to the 'cancel_button_click_event_handler' and 'clear_button_click_event_handler' methods, respectively. The class utilizes the 'ExecutionEngine' class to execute the simulation, displaying the results through the 'display_simulation_results' method. Overall, the RiskSimulator class encapsulates the functionality needed to orchestrate and present Monte Carlo simulations in an interactive manner.

Methods:

- __init__(self, max_runs=default_maximum_iterations): Constructor method initializing the user interface components, buttons, and output display for Monte Carlo Simulation. It sets up the 'output' widget, maximum iteration number, UI configurator, 'Run,' 'Exit,' and 'Clear' buttons, and the simulation progress widget.
- run_button_click_event_handler(self, button): Event handler method triggered when the 'Run' button is clicked. Performs the Monte Carlo simulation, displaying simulation results and handling errors.
- cancel_button_click_event_handler(self, button): Event handler method triggered when the 'Cancel' button is clicked. Sets the 'delay_process' attribute to False, stopping the Monte Carlo simulation process.
- clear_button_click_event_handler(self, button): Event handler method triggered when the 'Clear' button is clicked. Clears the output displayed in the user interface.
- execute(self): Executes the Monte Carlo simulation process. Enters a loop, continuously polls UI events, and sleeps to keep the UI responsive. After the simulation, sets the visibility of certain UI buttons to 'hidden' and displays a completion message.

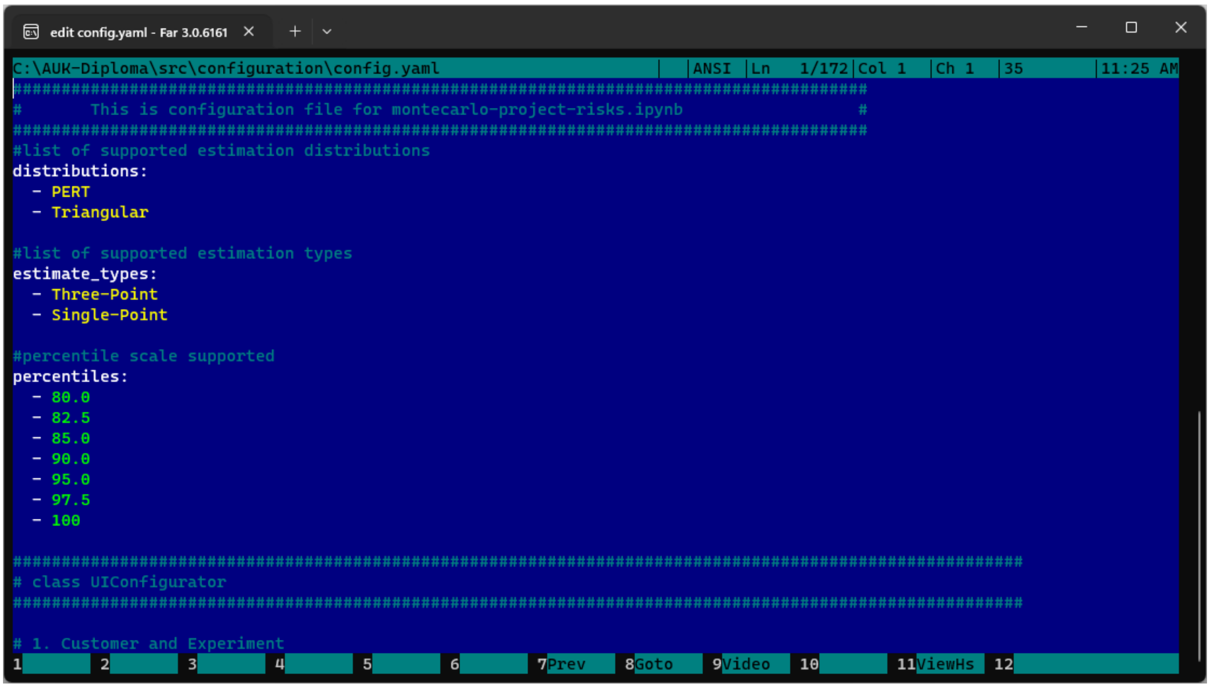
To summarize, the RiskSimulator simplifies the complexities of Monte Carlo simulations, making it accessible and efficient for users to analyze and understand project risks and uncertainties. Its seamless integration with the underlying components enhances user experience, offering a comprehensive tool for risk assessment and decision-making in project management.

4.5. YAML Configuration File

The *config.yaml* file serves as a configuration file for our Python-based project risk estimation and simulation tool, featuring distinct sections indicative of various underlying classes responsible for specific functionalities. By leveraging this configuration file, project managers and analysts gain granular control over the tool's functionalities, ensuring adaptability to diverse project scenarios.

YAML is a human-readable data serialization format that allows you to define configuration parameters in a clear, concise, and easily understandable manner. Its syntax is designed to be both human-friendly and machine-readable, making it an ideal choice for configuration files in various applications, including software engineering projects.

In the context of your Monte Carlo risk simulation library, using YAML to configure parameters such as project durations under uncertainty provides a structured and intuitive way to manage the complexities of the simulation process. The key-value pairs in YAML allow you to articulate a range of values and probabilities associated with each parameter, enabling a more



```
edit config.yaml - Far 3.0.6161
C:\AUK-Diploma\src\configuration\config.yaml        ANSI  Ln  1/172  Col 1  Ch 1  35  11:25 AM
#########################################################################################
#       This is configuration file for montecarlo-project-risks.ipynb                   #
#########################################################################################
#list of supported estimation distributions
distributions:
  - PERT
  - Triangular

#list of supported estimation types
estimate_types:
  - Three-Point
  - Single-Point

#percentile scale supported
percentiles:
  - 80.0
  - 82.5
  - 85.0
  - 90.0
  - 95.0
  - 97.5
  - 100

#########################################################################################################
# class UIConfigurator
#########################################################################################################

# 1. Customer and Experiment
1  2  3  4  5  6  7Prev  8Goto  9Video  10  11ViewHs  12
```

Fig. 7. YAML configuration file

realistic representation of uncertainties. Additionally, YAML supports nested structures, which can be advantageous when dealing with multi-dimensional configurations, ensuring a hierarchical organization that aligns with the intricacies of risk simulations.

The following table provides a clear overview of the different sections in config.yaml file and their respective descriptions.

Table 6. YAML configuration file section and their meaning

#	Section	Description
1	Distributions	Supported estimation distributions (e.g., PERT, Triangular).
2	Estimate Types	Supported estimation types (e.g., Three-Point, Single-Point).
3	Percentiles	Supported percentile scales (e.g., 80.0, 82.5, ..., 100).
4	UI Configurator	Configuration for the user interface elements and data mappings.
5	Global Project Risks	Configuration options for incorporating external project risks.
6	Process Parameters	Parameters for fine-tuning the simulation process and result export.
7	Risk Simulator & Execution Engine	Parameters for controlling simulation execution

4.6. Main code and Usage Example

The Monte Carlo simulation software provided by the monte_carlo_risk_project module offers a user-friendly experience for risk analysis in project management. With just a few lines of code, users can create an instance of the RiskSimulator class, which automatically initializes a graphical user interface (GUI) for configuring simulation parameters and initiating simulations. The simplicity of the software lies in its intuitive design, where users can execute Monte Carlo simulations seamlessly by invoking the execute() method. The software takes care of the complexities associated with risk assessment, providing real-time progress updates and presenting results in an easily understandable format. The straightforward integration and execution process make it effortless for users to leverage the power of Monte Carlo simulations, enabling efficient decision-making in the face of project uncertainties.

Screenshot below demonstrates simplicity for using our implementation of Monte Carlo Method

```
import monte_carlo_risk_project as mcrpr

simulator = mcrpr.RiskSimulator()
simulator.execute()
del simulator

```

Fig.8. Example of Programmatic Usage of Software

4.7. Working Environment

The working environment for our Monte Carlo simulator solution is intentionally designed to be versatile and lightweight, requiring minimal setup beyond a standard Jupyter Notebook installation. Our solution does not depend on any specific system configuration, making it accessible and easy to use for a wide range of users.

To utilize our Monte Carlo simulator, users need a Jupyter Notebook environment, which is a widely adopted platform for data analysis and scientific computing. Jupyter Notebook can be installed on various operating systems, including Windows, macOS, and Linux, ensuring compatibility with most computing environments.

Once Jupyter Notebook is installed, users can seamlessly import and execute our Monte Carlo simulator code, as demonstrated in the provided code snippets. The absence of complex dependencies or system-specific requirements simplifies the setup process and ensures that users can quickly start working with the simulator.

Furthermore, Jupyter Notebook's interactive nature allows users to experiment with different simulation parameters, visualize results, and document their findings in a collaborative and user-friendly manner. This working environment fosters flexibility, accessibility, and ease of use, making our Monte Carlo simulator solution accessible to a broad audience of data analysts, engineers, and decision-makers.

4.8. Installation Notes

Following steps have to be done to get our solution ready for using:

1. Clone the solution from git repository to your local directory

2. Copy the obtained directory structure to the root directory of your local Jupter Notebook

instance. For example, for computer running on Windows operating system this folder will be the folder C:\Users\YourUserDir

You can also upload the directory structure to any cloud version of Jupyter notebook, such as Google Collab - https://colab.research.google.com/. The structure of the solution with description of all files and folders are provided on Fig. 8 below:

```
--MSE22                              -->- root folder
    ¦   main.ipynb                   -->- Jupyter notebook with example of usage
    ¦   monte_carlo_risk_project.ipynb -->- library implementation in the format of Jupyer notebook
    ¦   monte_carlo_risk_project.py  -->- library implementation in the format of pure Python
    +---configuration                -->- configuration folder
    ¦       config.yaml              -->- configuration file
    +---input                        -->- input folder
    ¦   ¦   project-activities.xlsx  -->- project activities sample data
    ¦   ¦   project-risks.xlsx       -->- proejct risks samnple data
    +---output                       -->- folder for exporting calculations
```

Fig.9. File Structure of Software

4.9. Detailed Description of UI Parameters

Table 7. UI Parameters Description by Sections

Parameters Section	Parameter Name	Description
1.Customer and Experiment	Customer	Arbitrary string used only for structuring and proper documenting the results
	Experiment Name	Arbitrary string used only for structuring and proper documenting the results
2. Project Activities Data Mapping and Distribution	Activities Estimates Type	Drop Down box dedicated to selecting the activities estimate types available for given experiment. Options for drop-down box are defied in config.yaml file
	Activities Data File	Editable field which indicates name of Excel file with project activities data. File should be locate in the /input folder
	Activities Sheet	Editable field which indicates name of Excel sheet in the file with project activities data.
	Activities Data Range	Editable field which indicates data range on Excel sheet in the file with project activities data.
	Activity Column #1	Column in Excel file specifying any categorical / descriptive attribute of project activities. Needed to help identify activities after the import. Values example: A, B,C,…. Should be selected from drop-down-box.

	Activity Column #2	Column in Excel file specifying any categorical / descriptive attribute of project activities. Needed to help identify activities after the import. Values example: A, B,C,…. Should be selected from drop-down-box.
	Activity Column #3	Column in Excel file specifying any categorical / descriptive attribute of project activities. Needed to help identify activities after the import. Values example: A, B,C,…. Should be selected from drop-down-box.
	Activity Column #4	Column in Excel file specifying any categorical / descriptive attribute of project activities. Needed to help identify activities after the import. Values example: A, B,C,…. Should be selected from drop-down-box.
	Activities Distribution	Drop Down box dedicated to selecting the random variable distribution type for project activities. Values example: A, B, C, …. Should be selected from drop-down-box. Specific values are defined in the configuration file. Typical values are PERT, Triangular
	Most-Likely Estimate Column	Column in Excel file specifying Most-Likely estimate for project activities in case of Three-Point estimate type selected. If Singe-Point selected than this is just the estimate. Values example: A, B, C, …. Should be selected from drop-down-box.
	Optimistic Estimate Column	Column in Excel file specifying Optimistic estimate for project activities in case of Three-Point estimate type selected. If Singe-Point selected than this is just the estimate. Values example: A, B, C, …. Should be selected from drop-down-box
	Pessimistic Estimate Column	Column in Excel file specifying Pessimistic estimate for project activities in case of Three-Point estimate type selected. If Singe-Point selected than this is just the estimate. Values example: A, B, C, …. Should be selected from drop-down-box
3. Global Project Risks Data Mapping and Distribution	Include Global Project Risks	Checkbox indicating whether user wants to include genera risks information
	Risks Data File	Editable field which indicates name of Excel file with project risks data. File should be locate in the /input folder

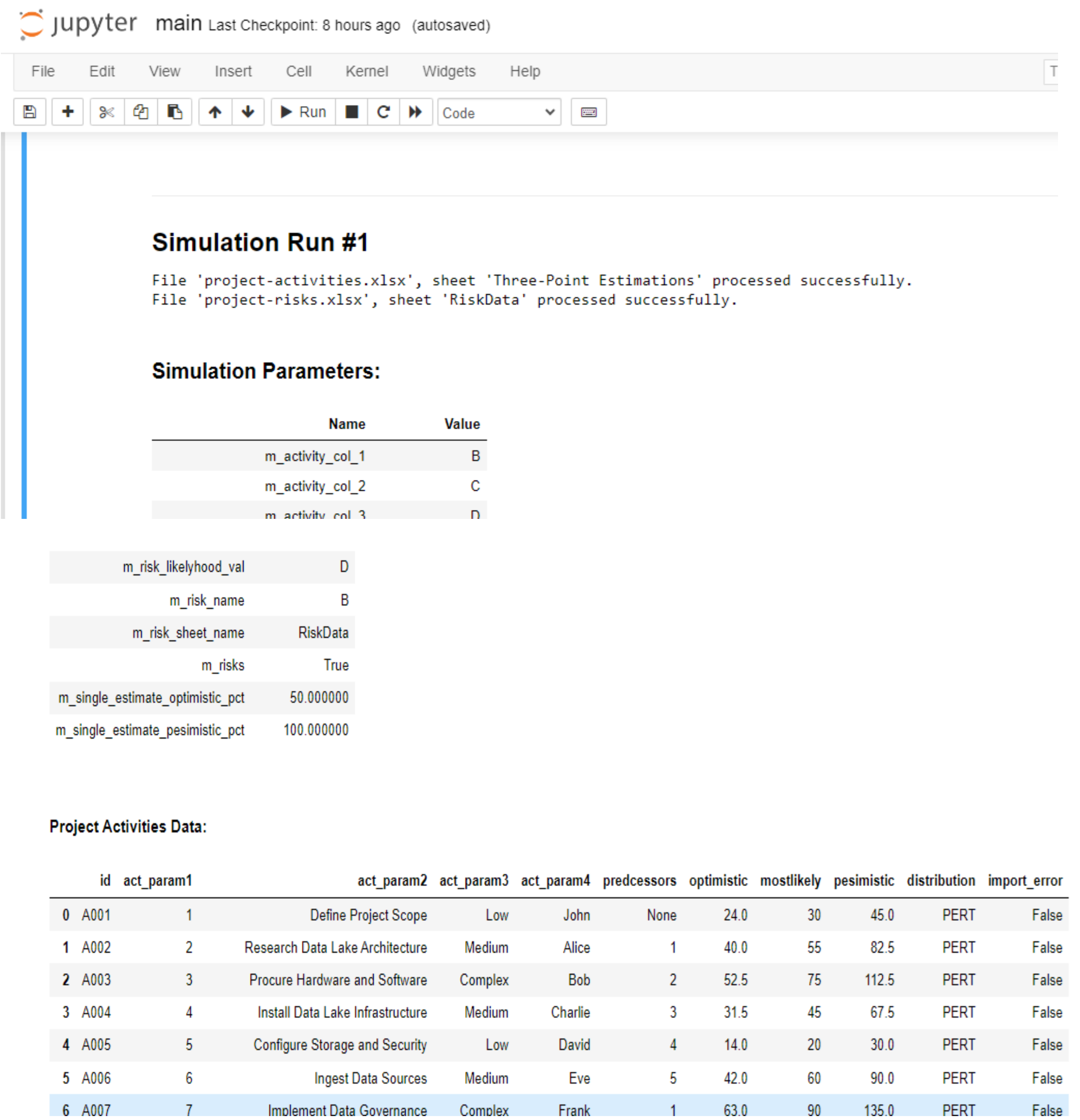
	Risk Data Sheet Name	Editable field which indicates name of Excel sheet in the file with project risk data.
	Risk Data Range	Editable field which indicates name of Excel sheet in the file with project risk data.
	Global Risks Distribution	Drop Down box dedicated to selecting the random variable distribution type for project risks. Values example: A, B, C, …. Should be selected from drop-down-box. Specific values are defined in the configuration file. Typical values are: PERT
	Risk ID Column	Column in Excel file specifying appropriate categorical / descriptive attribute of project risk dataset. Needed to map/identify elements of risk data. Values example: A, B,C…. Should be selected from drop-down-box.
	Risk Name Column	Column in Excel file specifying appropriate categorical / descriptive attribute of project risk dataset. Needed to map/identify elements of risk data. Values example: A, B,C…. Should be selected from drop-down-box.
	Likelihood Column	Column in Excel file specifying appropriate categorical / descriptive attribute of project risk dataset. Needed to map/identify elements of risk data. Values example: A, B,C…. Should be selected from drop-down-box.
	Likelihood % Column	Column in Excel file specifying appropriate categorical / descriptive attribute of project risk dataset. Needed to map/identify elements of risk data. Values example: A, B,C…. Should be selected from drop-down-box.
	Optimistic Impact Column	Column in Excel file specifying appropriate categorical / descriptive attribute of project risk dataset. Needed to map/identify elements of risk data. Values example: A, B,C…. Should be selected from drop-down-box.
	Optimistic Impact % Column	Column in Excel file specifying appropriate categorical / descriptive attribute of project risk dataset. Needed to map/identify elements of risk data. Values example: A, B,C…. Should be selected from drop-down-box.

	Most-likely Impact Column	Column in Excel file specifying appropriate categorical / descriptive attribute of project risk dataset. Needed to map/identify elements of risk data. Values example: A, B,C…. Should be selected from drop-down-box.
	Most-likely Impact % Column	Column in Excel file specifying appropriate categorical / descriptive attribute of project risk dataset. Needed to map/identify elements of risk data. Values example: A, B,C…. Should be selected from drop-down-box.
	Pessimistic Impact Column	Column in Excel file specifying appropriate categorical / descriptive attribute of project risk dataset. Needed to map/identify elements of risk data. Values example: A, B,C…. Should be selected from drop-down-box.
4. Process Parameters	Number of Iterations	Slider control defining the number of simulation iterations
	Export results	Checkbox indicating whether user wants t export internal calculations into external file in Excel format in /output folder
	High Risk %	Probability of estimation in % form for High-Risk scenario. Percent value is defined in configuration file
	Moderate Risk %	Probability of estimation in % form for Moderate Risk scenario. Percent value is defined in configuration file
	Low Risk %	Probability of estimation in % form for Low Risk scenario. Percent value is defined in configuration file

4.10 System Outputs

4.10.1 Simulation Parameters and Imported Data

This output provides a detailed overview of the simulation's setup. It includes information on the simulation parameters that users can customize, such as the number of iterations, risk factors, and project activities. Additionally, it covers the process of importing project data, which serves as the foundation for running simulations. Understanding these parameters and imported data is crucial as they lay the groundwork for the subsequent analysis and risk assessment. Example of this output is provided on the screenshot below:



Simulation Run #1

```
File 'project-activities.xlsx', sheet 'Three-Point Estimations' processed successfully.
File 'project-risks.xlsx', sheet 'RiskData' processed successfully.
```

Simulation Parameters:

Name	Value
m_activity_col_1	B
m_activity_col_2	C
m_activity_col_3	D
m_risk_likelyhood_val	D
m_risk_name	B
m_risk_sheet_name	RiskData
m_risks	True
m_single_estimate_optimistic_pct	50.000000
m_single_estimate_pesimistic_pct	100.000000

Project Activities Data:

	id	act_param1	act_param2	act_param3	act_param4	predcessors	optimistic	mostlikely	pesimistic	distribution	import_error
0	A001	1	Define Project Scope	Low	John	None	24.0	30	45.0	PERT	False
1	A002	2	Research Data Lake Architecture	Medium	Alice	1	40.0	55	82.5	PERT	False
2	A003	3	Procure Hardware and Software	Complex	Bob	2	52.5	75	112.5	PERT	False
3	A004	4	Install Data Lake Infrastructure	Medium	Charlie	3	31.5	45	67.5	PERT	False
4	A005	5	Configure Storage and Security	Low	David	4	14.0	20	30.0	PERT	False
5	A006	6	Ingest Data Sources	Medium	Eve	5	42.0	60	90.0	PERT	False
6	A007	7	Implement Data Governance	Complex	Frank	1	63.0	90	135.0	PERT	False

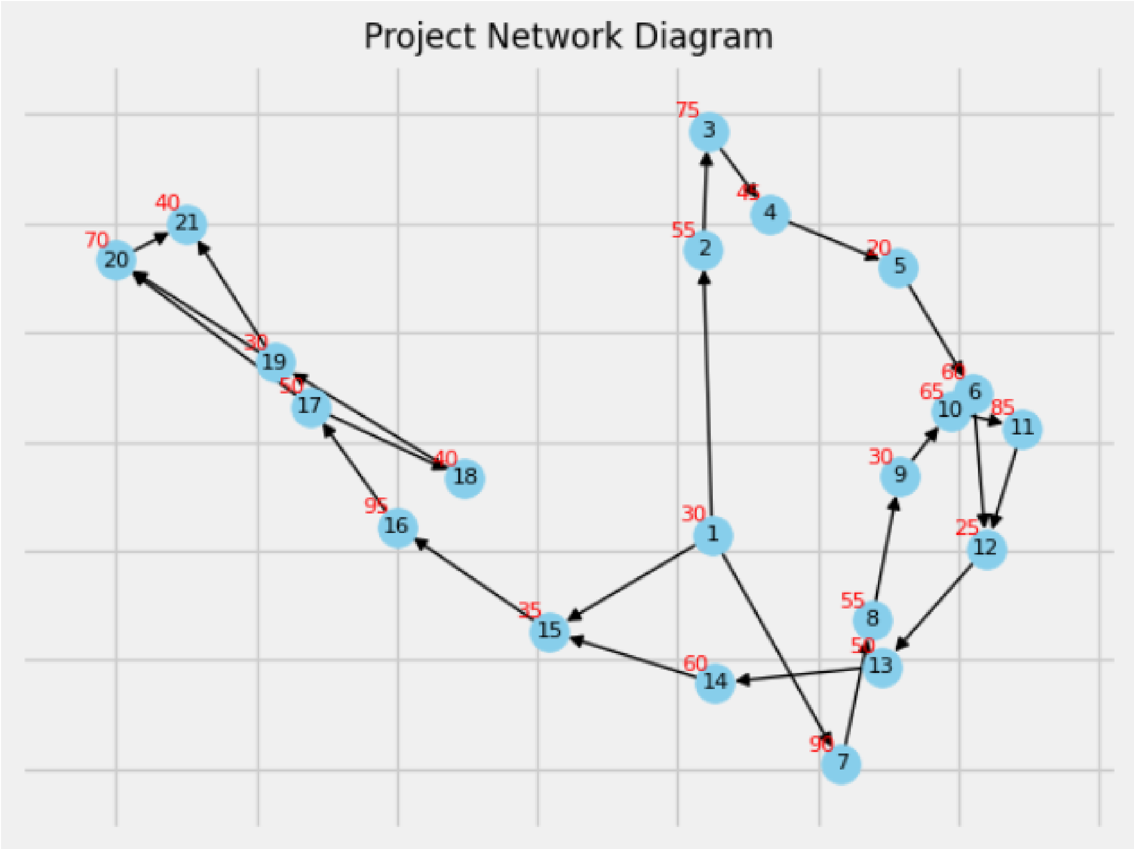
Fig. 10. Simulation Parameters and Imported Data

.

4.10.2 CPM Project Network Diagram

The CPM Project Network Diagram visually represents the interdependencies between various project activities. It showcases the critical path, highlighting the sequence of activities that collectively determine the project's duration. This diagram aids users in comprehending the project's structure and identifying critical activities that may impact the overall timeline.

It's an invaluable tool for project managers and stakeholders seeking to optimize project scheduling and resource allocation.




```
Calculated Project Duration (Critical Path Method): 780 days
```

Fig. 11. CMP Project network diagram.

4.10.3 Cumulative distribution and density functions chart

By adding this output, we delve into the Cumulative Distribution and Density Functions Chart, a fundamental element of Monte Carlo simulations. This chart provides insights into the range of possible project durations and the likelihood of each duration occurring. The cumulative distribution function (CDF) represents the probability of the project completing within a specific timeframe, while the density function (PDF) visualizes the probability density at each point. This chart enables decision-makers to assess the risk associated with different project completion times and make informed choices.

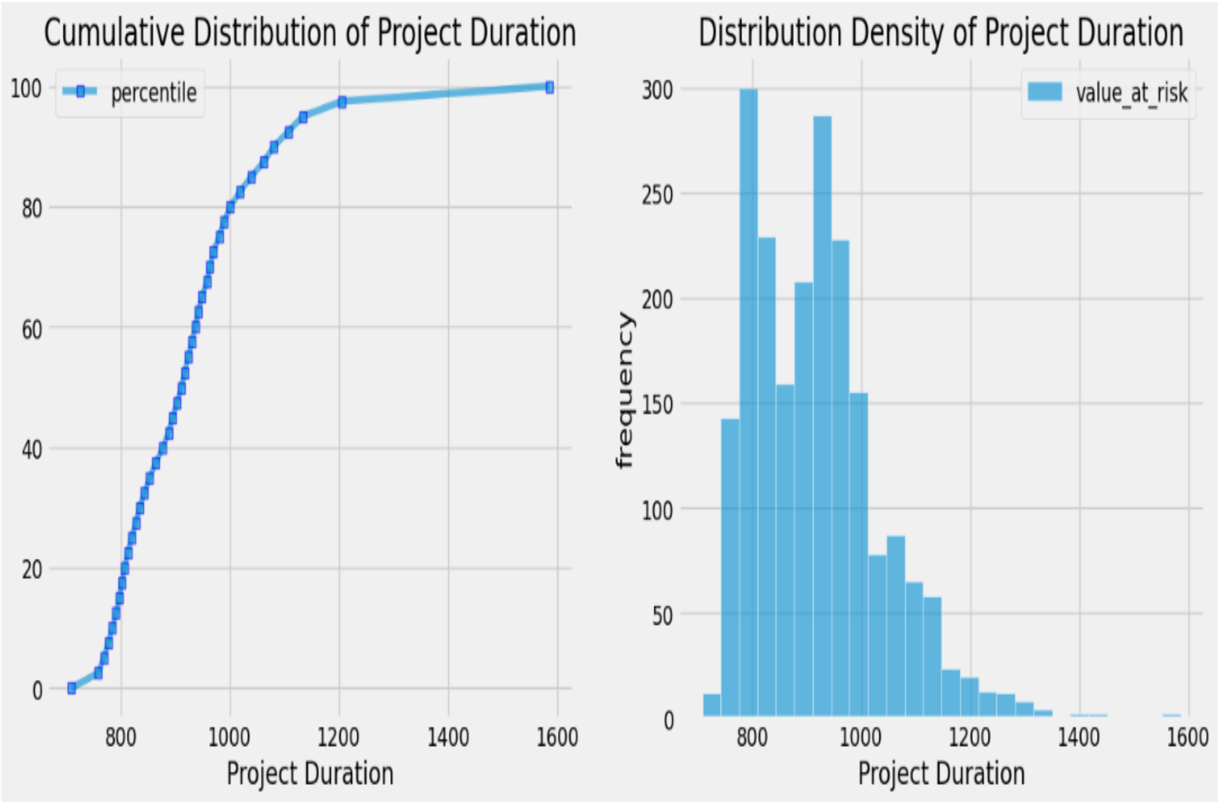


Fig. 12. Distribution and Density Functions

4.10.4 Tornado Chart

The Tornado Chart, a central component of the Monte Carlo simulation documentation, plays a pivotal role in risk analysis and management. This chart is named after its distinctive appearance, where bars representing different risk factors resemble the swirling winds of a tornado. Its purpose is to unveil the vulnerabilities and strengths within a project by highlighting the factors that have the most substantial impact on project duration.

The Tornado Chart provides a concise yet comprehensive overview of how individual risk factors can influence project outcomes. Each bar in the chart represents a specific risk factor, and the height of the bar signifies the magnitude of its impact. The taller the bar, the more significant the effect of that particular factor on project duration.

Project managers and stakeholders rely on the Tornado Chart to prioritize their risk mitigation efforts effectively. By identifying and focusing on the top-ranking risk factors, they can allocate resources and attention to those aspects of the project that are most likely to lead to deviations from the baseline schedule.

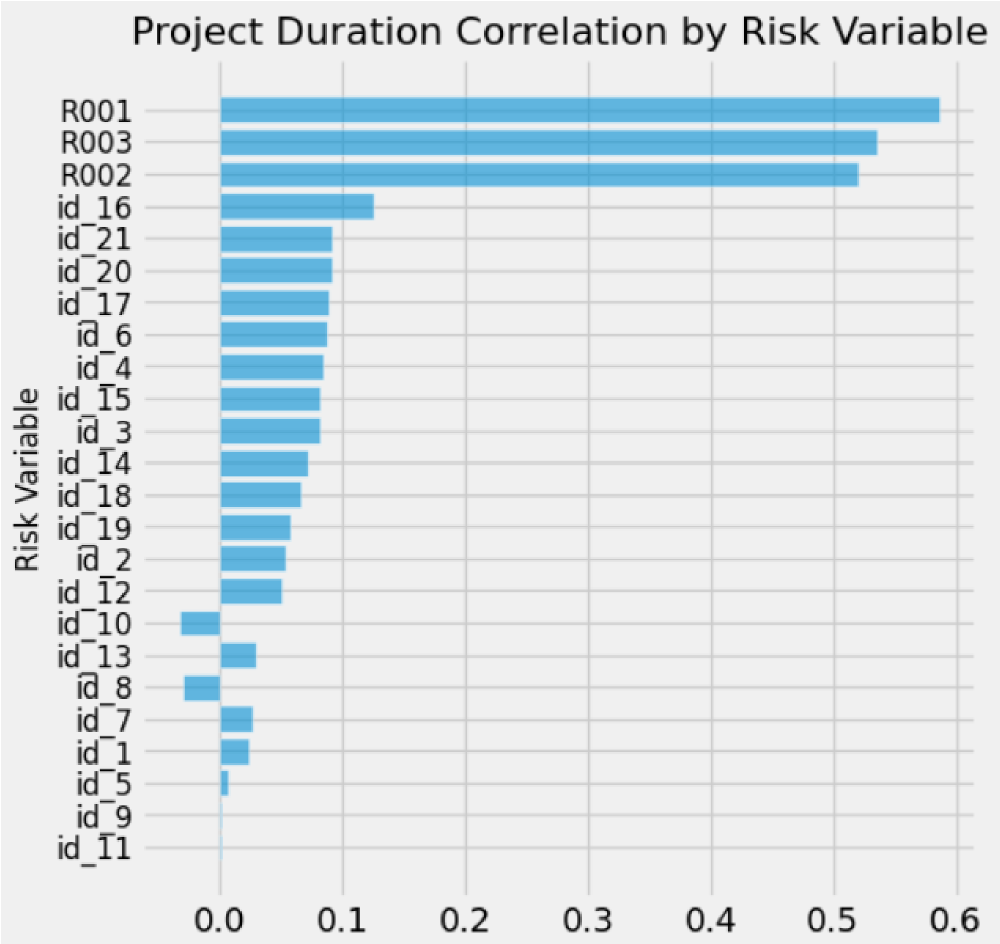


Fig. 13. Tornado Chart, correlation between Project Duration and Individual Risks

4.10.5 Project Duration Estimates by Scenario.

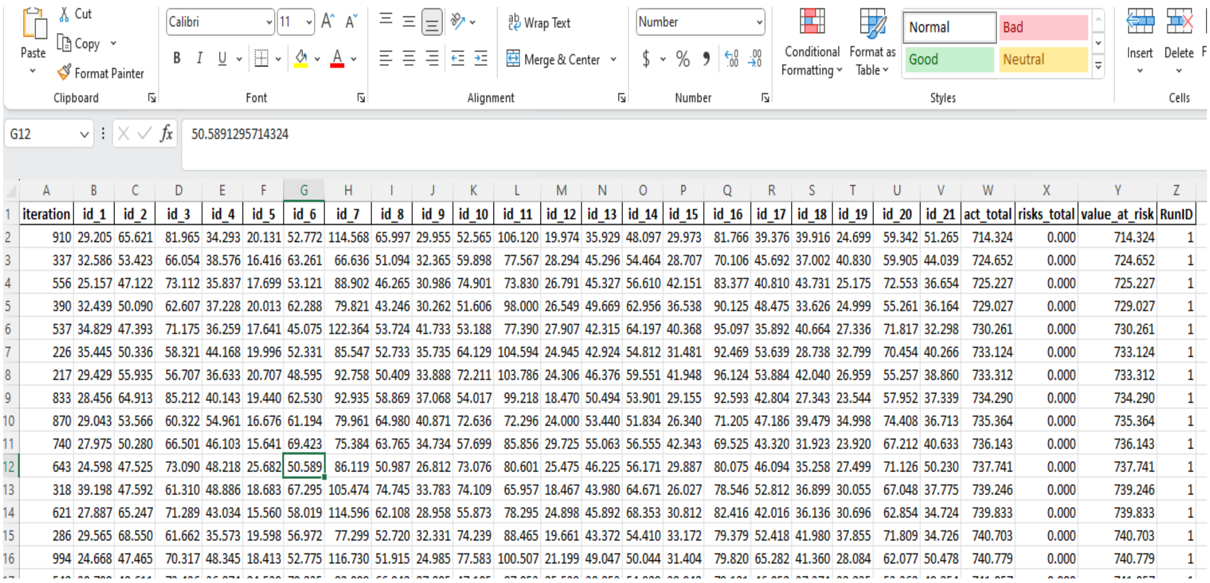
This section presents the results of the Monte Carlo simulations in the form of Project Duration Estimates by Scenario. It provides users with a comprehensive view of how project duration can vary under different conditions and risk scenarios. By exploring these estimates, stakeholders gain a deeper understanding of the potential outcomes and uncertainties associated with the project. This information empowers them to make proactive decisions, allocate resources effectively, and develop contingency plans based on the identified scenarios.

Project Duration Estimates:

Risk Scenario	Probability	Effort Estimate	Explanation
High Risk	80.000000	1000.890000	Project Duration value is less or equal to 1000.89 with probability of 80.0 %
Moderate Risk	90.000000	1082.320000	Project Duration value is less or equal to 1082.32 with probability of 90.0 %
Low Risk	97.500000	1206.290000	Project Duration value is less or equal to 1206.29 with probability of 97.5 %

Fig. 14. Resulting Project Duration Estimate by Scenarios

4.10.6 Detailed Calculations Export File.



iteration	id_1	id_2	id_3	id_4	id_5	id_6	id_7	id_8	id_9	id_10	id_11	id_12	id_13	id_14	id_15	id_16	id_17	id_18	id_19	id_20	id_21	act_total	risks_total	value_at_risk	RunID
910	29.205	65.621	81.965	34.293	20.131	52.772	114.568	65.997	29.955	52.565	106.120	19.974	35.929	48.097	29.973	81.766	39.376	39.916	24.699	59.342	51.265	714.324	0.000	714.324	1
337	32.586	53.423	66.054	38.576	16.416	63.261	66.636	51.094	32.365	59.898	77.567	28.294	45.296	54.464	28.707	70.106	45.692	37.002	40.830	59.905	44.039	724.652	0.000	724.652	1
556	25.157	47.122	73.112	35.837	17.699	53.121	88.902	46.265	30.986	74.901	73.830	26.791	45.327	56.610	42.151	83.377	40.810	43.731	25.175	72.553	36.654	725.227	0.000	725.227	1
390	32.439	50.090	62.607	37.228	20.013	62.288	79.821	43.246	30.262	51.606	98.000	26.549	49.669	62.956	36.538	90.125	48.475	33.626	24.999	55.261	36.164	729.027	0.000	729.027	1
537	34.829	47.393	71.175	36.259	17.641	45.075	122.364	53.724	41.733	53.188	77.390	27.907	42.315	64.197	40.368	95.097	35.892	40.664	27.336	71.817	32.298	730.261	0.000	730.261	1
226	35.445	50.336	58.321	44.168	19.996	52.331	85.547	52.733	35.735	64.129	104.594	24.945	42.924	54.812	31.481	92.469	53.639	28.738	32.799	70.454	40.266	733.124	0.000	733.124	1
217	29.429	55.935	56.707	36.633	20.707	48.595	92.758	50.409	33.888	72.211	103.786	24.306	46.376	59.551	41.948	96.124	53.884	42.040	26.959	55.257	38.860	733.312	0.000	733.312	1
833	28.456	64.913	85.212	40.143	19.440	62.530	92.935	58.869	37.068	54.017	99.218	18.470	50.494	53.901	29.155	92.593	42.804	27.343	23.544	57.952	37.339	734.290	0.000	734.290	1
870	29.043	53.566	60.322	54.961	16.676	61.194	79.961	64.980	40.871	72.636	72.296	24.000	53.440	51.834	26.340	71.205	47.186	39.479	34.998	74.408	36.713	735.364	0.000	735.364	1
740	27.975	50.280	66.501	46.103	15.641	69.423	75.384	63.765	34.734	57.699	85.856	29.725	55.063	56.555	42.343	69.525	43.320	31.923	23.920	67.212	40.633	736.143	0.000	736.143	1
643	24.598	47.525	73.090	48.218	25.682	50.589	86.119	50.987	26.812	73.076	80.601	25.475	46.225	56.171	29.887	80.075	46.094	35.258	27.499	71.126	50.230	737.741	0.000	737.741	1
318	39.198	47.592	61.310	48.886	18.683	67.295	105.474	74.745	33.783	74.109	65.957	18.467	43.980	64.671	26.027	78.546	52.812	36.899	30.055	67.048	37.775	739.246	0.000	739.246	1
621	27.887	65.247	71.289	43.034	15.560	58.019	114.596	62.108	28.958	55.873	78.295	24.898	45.892	68.353	30.812	82.416	42.016	36.136	30.696	62.854	34.724	739.833	0.000	739.833	1
286	29.565	68.550	61.662	35.573	19.598	56.972	77.299	52.720	32.331	74.239	88.465	19.661	43.372	54.410	33.172	79.379	52.418	41.980	37.855	71.809	34.726	740.703	0.000	740.703	1
994	24.668	47.465	70.317	48.345	18.413	52.775	116.730	51.915	24.985	77.583	100.507	21.199	49.047	50.044	31.404	79.820	65.282	41.360	28.084	62.077	50.478	740.779	0.000	740.779	1

Fig. 15. Calculations Export In Excel Format

This is the Excel file with detailed calculations which is being generated in /Output folder. File could be used for troubleshooting or conducting some advanced analysis.

5 Conclusions

The development and implementation of our custom Monte Carlo Risk Simulator have resulted in a powerful and flexible tool for risk analysis and decision-making in various domains. Throughout this documentation, we have discussed the architecture, design, and key features of the simulator. In this section, we summarize our findings and emphasize the benefits, drawbacks, and future prospects of our solution.

5.10 Benefits

1. Simplicity of Use

One of the standout advantages of our simulator is its user-friendliness. The intuitive user interface, coupled with clear documentation, ensures that users, even those without extensive technical backgrounds, can readily employ the tool. This simplicity enables a broader range of professionals to harness the power of Monte Carlo simulations for risk assessment.

2. Full Control on Calculations

Our solution provides users with full control over simulation parameters, enabling them to customize simulations to match specific scenarios accurately. From adjusting the number of iterations to modifying input data, users have the flexibility to tailor simulations to their unique needs. This control fosters a deep understanding of risk factors and empowers users to make informed decisions.

3. Transparency and Insight

The architecture of the simulator is designed to be transparent. The Component Diagram (Level 3) reveals the internal workings of the simulator, making it accessible for those interested in understanding the underlying calculations. This transparency promotes trust in the results and encourages collaboration among teams.

4. Extensibility

Our simulator is built with extensibility in mind. Users can add custom functionality or integrate it with other tools and systems as needed. This extensibility allows for the adaptation of the simulator to various domains beyond its initial scope, making it a versatile asset for organizations.

5.11 Drawbacks and Limitation

1. Testing Complexity

One of the significant challenges of our custom simulator is the thorough testing required to ensure accuracy and reliability. Monte Carlo simulations inherently involve randomness, making it essential to conduct extensive testing to validate results. This testing phase can be time-consuming and resource intensive.

2. Limited number of distributions.

For now, we implemented only PERT and Triangular distributions for activities and PERT for global risks. We may need implement a wider spectrum of distributions to better reflect different practical cases.

5.12 Future Improvements

Despite the success of our current implementation, there are several avenues for future improvements:

1. Enhanced Visualization

Integrating advanced visualization features into the simulator can enhance its usability further. Visual representations of simulation results, such as charts and graphs, can provide users with a more intuitive understanding of risk scenarios.

2. Performance Optimization

Continued efforts to optimize the simulator's performance, particularly for large-scale simulations, can lead to faster and more efficient risk assessments. Utilizing parallel processing and distributed computing techniques can be explored.

3. Integration with Data Sources

Enabling seamless integration with external data sources, such as databases and APIs, can enhance the simulator's capabilities. This would allow users to access real-time data for simulations, increasing the accuracy of risk assessments.

4. Scenario Management

Developing a scenario management system within the simulator can streamline the process of comparing different risk scenarios. Users can save and revisit specific simulation setups, facilitating scenario analysis and comparison.

In conclusion, our custom Monte Carlo Risk Simulator stands as a valuable tool for organizations seeking to navigate uncertain environments and make informed decisions. Its simplicity of use, transparency, and extensibility make it a versatile asset, while ongoing improvements can further enhance its capabilities. Despite the challenge of testing, the benefits of full control over calculations and the insights gained justify its implementation. As organizations continue to face complex and dynamic risks, our simulator remains a reliable ally in their risk management toolkit.

References

1. PMI, Success Rates Rise 2017. "9th Global Project Management Survey," PMI's Pulse Prof., p. 32, 2017.
2. Project Management Institute (PMI), "Success in Disruptive Times," Pulse Prof., vol. 10th Globa, p. 35, 2018.
3. L. A. Kappelman, R. McKeeman, and L. Zhang, "Early warning signs of it project failure: the dominant Dozen," Edpacs, vol. 35, no. 1, pp. 1–10, 2007.
4. PMI (2020). Ahead of the Curve: Forging a Future-Focused Culture. *Pulse of the Profession*.
5. PMI (2021). Beyond Agility. *Pulse of the Profession.*
6. Tiffani Iacolino. (2022). *Business analysis tips for avoiding failure rates, part 1: Analyst catalyst blog*. Tips for avoiding failure rates for data analytics projects | IIBA®. https://www.iiba.org/business-analysis-blogs/business-analysis-tips-for-avoiding-failure-rates-part-1/. Accessed: Dec 10, 2023.
7. Project Management Institute. (2013). *Software Extension to PMBOK® Guide (5th ed.*). Project Management Institute.
8. Project Management Institute. (2017). *A guide to the Project Management Body of Knowledge (PMBOK guide)* (6th ed.). Project Management Institute.
9. Project Management Institute. (2021). *A guide to the Project Management Body of Knowledge (PMBOK guide)* (7th ed.). Project Management Institute.
10. Wikimedia Foundation. (2023, March 24). *Triangular distribution*. Wikipedia. https://en.wikipedia.org/wiki/Triangular_distribution
11. Wikimedia Foundation. (2022, July 24). *Pert distribution*. Wikipedia. https://en.wikipedia.org/wiki/PERT_distribution
12. Avlijas, G. (2019). Examining the Value of Monte Carlo Simulation for Project Time Management. *Management:Journal Of Sustainable Business And Management Solutions In Emerging Economies, 24*(1), 11-23. doi:10.7595/management.fon.2018.0004
13. Kroese, D. P., Taimre, T., & Botev, Z. I. (2014). Handbook of Monte Carlo Methods. John Wiley & Sons.

14. Eckhardt, R. (1987). Stanislaw Ulam, John von Neumann, and the Monte Carlo method. Los Alamos Science, 15, 131-137.
15. Project Management Institute. (2017). A Guide to the Project Management Body of Knowledge (PMBOK Guide). Project Management Institute.
16. Kwak, Y. H., & Ingall, L. (2007). Exploring Monte Carlo simulation applications for risk analysis of time and cost for complex infrastructure systems. Journal of Infrastructure Systems, 13(1), 3-12.
17. Acebes, F., Pellicer, E., & López-Paredes, A. (2015). A Monte Carlo simulation-based risk model for project scheduling with uncertainty. Journal of Computational and Applied Mathematics, 275, 24-32.
18. Hazir, Ö. (2015). Project time-risk analysis with Monte Carlo simulation. International Journal of Project Management, 33(2), 523-532.
19. Williams, T. M. (2003). The need for new paradigms for complex projects. International Journal of Project Management, 21(2), 71-81.
20. Salkeld, W. J. (2016). Monte Carlo risk analysis in project planning. Project Management Journal, 47(2), 47-58.
21. Vanhoucke, M. (2016). Integrated project management and control: First comes the theory, then the practice. International Journal of Project Management, 34(7), 1046-1059.
22. Wanner, R. A. (2013). A comparison of Monte Carlo simulation methods for project management. Decision Analysis, 10(4), 317-333.
23. Meredith, J. R., & Mantel, S. J. (2011). Project Management: A Managerial Approach. John Wiley & Sons.
24. Monte Carlo simulation in Crystal Ball 7 - Researchgate. (n.d.). URL: https://www.researchgate.net/profile/Mohamed-Mourad-Lafifi/post/How_can_I_make_optimization_using_SPSS_or_other_programs_to_select_the_best_samples_that_fulfill_the_standard/attachment/59d6497579197b80779a3f58/AS%3A471314843607040%401489381623169/download/Monte+Carlo+Simulation+in+Crystal_Ball_7.3_UK.pdf. Accessed 10 Jan. 2024.

25. Intaver Institute Inc.(2023) Monte Carlo Simulation with Microsoft Project, Project Decision and Risk Analysis Journal, https://intaver.org/Articles/RP_Art_MSProjectRiskAnalysis2.html . Accessed 10 Jan. 2024.

26. Jacques Alexis (2023). Risk Analytics with Primavera Risk Analysis. Management Blog. URL: Risk Analytics with Primavera Risk Analysis. - YouTube. Accessed 10 Jan. 2024
27. Alfasoft (Mar. 2023). @Risk - Monte Carlo Simulation Analysis in Excel. URL: https://alfasoft.com/software/statistics-and-data-analysis/risk-optimization-and-quality-analysis/risk/
28. Vose Software (2023) Tamara - Project Risk Analysis Software. URL: Project Risk Analysis Software for Primavera and MS Project | Vose Software Accessed 10 Jan. 2024.
29. *Safran risk: Risk analytics solutions: United Kingdom*. Safran. (n.d.). URL: https://www.safran.com/en-gb/risk-analytics-solutions. Accessed 10 Jan. 2024.
30. ModelRisk: FREE Risk Modelling within Microsoft Excel. URL: https://riskmanagementguru.com/modelrisk-free-risk-modelling-within-microsoft-excel.html. Accessed 10 Jan. 2024
31. S.Atin, R.Lubis.(2019). Implementation of Critical Path Method in Project Planning and Scheduling. IOP Conf. Series: Materials Science and Engineering 662 (2019) 022031. IOP Publishing. doi:10.1088/1757-899X/662/2/022031
32. Aliyu A.M.(2012). Project Management Using Critical Path Method (CPM):a pragmatic story. Global journal of pure and applied sciences vol. 18, no. 3&4, 2012: 197-206. DOI: http://dx.doi.org/10.4314/gjpas.v18i3-4.11. pp.197-206.
33. The C4 model for visualizing software architecture. URL: The C4 model for visualising software architecture. Accessed 10 Jan. 2024