

# Strategic Metrics for Risk Management Across Software Development Projects

## GROUP 2 MEMBERS:

- ~ LOKESH PUROHIT
- ~ MALIKA HAFIZA PASHA
- ~ MUSTAFA QURAISHI
- ~ SAJEEL MOHAMMED ABDUL
- ~ SHOAIBUDDIN MOHAMMED



# Table of Contents

Goal

Motivation

Introduction

- Challenges in Software Development
- Importance of Metrics

Methodology and Frameworks

- Project Risk Metric Methodology
- Enhanced Risk Assessment Framework for Software
- Systematic Methodology for Implementing Metrics Programs
- Framework for Continuous Improvement of Software Processes

Key Findings

Conclusion

Reference

# Goal

- ▶ The goal of this presentation is to outline a cohesive strategy for assessing and managing risks across multiple software development projects, utilizing a variety of metrics to ensure project success and mitigate potential failures.





# Motivation

## **1. Industry Context:**

Software projects are becoming more complex with higher risk levels, impacting successful outcomes.

## **2. Need for Improvement:**

High incidence of project failures and cost overruns highlights the need for better risk assessment and management strategies.

## **3. Advancement in Metrics:**

New developments in risk metrics provide improved tools and frameworks, enhancing risk visibility and management.

# Introduction





# Challenges in Software Development

## ► **1. Scope Creep:**

Expansion of project scope without corresponding adjustments in resources or timelines, leading to misalignment and potential failure.

## ► **2. Budget Overruns:**

Costs exceed initial estimates due to mismanagement or unforeseen issues, risking financial stability and necessitating scope reduction.

## ► **3. Delays:**

Projects not completed on schedule due to poor estimation or resource issues, impacting ROI and market presence.





# Importance of Metrics

▶ **1. Quantitative Analysis:**

Use data-driven KPIs and indexes (CPI, SPI) to assess project health and efficiency.

▶ **2. Risk Assessment:**

Identify and mitigate risks early through continuous metric monitoring to adjust scope or resources.

▶ **3. Performance Benchmarking:**

Compare project performance against historical or industry standards to drive improvements.

▶ **4. Resource Allocation:**

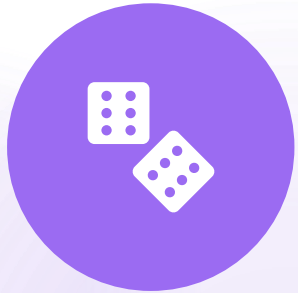
Optimize resource use to prevent bottlenecks and maximize efficiency.

# Methodology and Frameworks



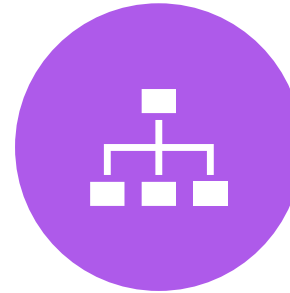


# 1. Project Risk Metric Methodology



## **Risk Measurement:**

Use the probability-impact product (PxV) to identify project risks early.

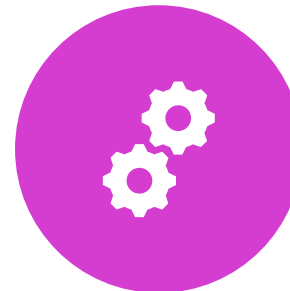


**Risk Categorization:** Classify and normalize risks on a scale of 1 to 5 based on historical data.



## **Analysis:**

Merge qualitative and quantitative data to refine risk predictions.



## **Integration:**

Adapt the metric for use within standard PMBOK risk management processes.

## 2. Enhanced Risk Assessment Framework for Software

01

**Structure:**

Manages risks in complex, multi-project environments.

02

**Indicators:**

Utilizes specific metrics like project size and resource allocation to assess risk.

03

**Scalability:**

Adapts to different organizational sizes and project scales.

04

**Feedback:**

Incorporates ongoing updates to risk assessments from new data.

### 3. Systematic Methodology for Implementing Metrics Programs

#### **Metric-Based Risk Identification:**

Employs metrics such as code complexity and performance benchmarks to assess project health.

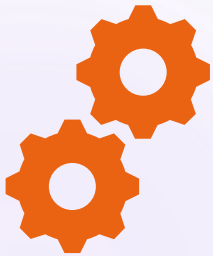
#### **Iterative Assessment:**

Performs frequent risk evaluations to respond to changes and external influences.

#### **Comprehensive Risk Drivers:**

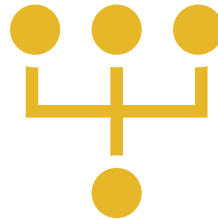
Addresses technical (e.g., bugs, requirement shifts) and organizational (e.g., resource and stakeholder issues) risks.

## 4. Framework for Continuous Improvement of Software Processes



### **Continuous Monitoring:**

Integrates regular risk assessments into daily operations.



### **Process-Oriented Risk Management:**

Uses metrics to enhance processes and reduce risks.



### **Organizational Learning:**

Leverages past data to refine risk strategies and foster improvement.





# Key Findings

## **1. Risk Metric Introduction:**

A normalized metric  $R=P \times V$  [Probability (P) and Outcome (V)] to enhance project uncertainty visibility and guide decisions on project trajectories.

## **2. Structured Risk Management:**

Adheres to PMBOK's comprehensive processes: planning, identification, analysis, response planning, and monitoring.

## **3. Enhanced Risk Framework:**

Develops a systematic framework for multi-project environments, focusing on risk categorization and prioritization.

## **4. Utility of Risk Metrics:**

Provides quick project comparisons but needs additional strategies for effective risk mitigation.

## **5. Systematic Integration:**

Integrates risk management within project management to systematically address project risks.

## **6. Application Across Projects:**

Uses "Risk Points" to facilitate decision-making and risk monitoring in multi-project settings.

# Conclusion

1

Focus on developing standardized risk metrics and models for managing risks in software projects, especially where resources are shared across multiple projects.

2

Advocate for integrated frameworks that combine risk and project management, emphasizing early, continuous risk assessment using both quantitative and qualitative methods.

3

Key challenges include the complexity of implementing these frameworks and the necessity for extensive training and sophisticated tools.

4

The overall recommendation is towards empirically validated practices that enhance project success and align with organizational objectives.

# References

[1] R. W. Ferguson, "A Project Risk Metric"

Link:<https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=ee5645aa1a30272a59c67d6a9220cfa8603ee272>

[2] T. Hall, N. Fenton, "Implementing Effective Software Metrics Programs"

Link:[https://www.researchgate.net/publication/3247256\\_Implementing\\_effective\\_software\\_metric\\_programs](https://www.researchgate.net/publication/3247256_Implementing_effective_software_metric_programs)

[3] N. R. Darwish, A. A. M. Warad, "An Enhanced Risk Assessment Framework for Software"

Link:[https://www.researchgate.net/publication/327105654\\_An\\_Enhanced\\_Risk\\_Assessment\\_Framework\\_for\\_Software\\_Development\\_in\\_Multiple\\_Projects\\_Environment](https://www.researchgate.net/publication/327105654_An_Enhanced_Risk_Assessment_Framework_for_Software_Development_in_Multiple_Projects_Environment)

[4] A. I. Rana, K. Saeed, U. Saeed, "Improving Software Processes using Metrics Plan"

Link:[https://www.academia.edu/580429/Improving\\_Software\\_Processes\\_using\\_Metrics\\_Plan](https://www.academia.edu/580429/Improving_Software_Processes_using_Metrics_Plan)



Thank You!