Journal of Software Engineering and Applications, 2024, 17, 228-245
https://www.scirp.org/journal/jsea
ISSN Online: 1945-3124
ISSN Print: 1945-3116

A Proposed Approach for Measuring Maturity Level of Software Delivery

Osama Hassan Mansour, Atef Raslan, Nagy Ramadan

Faculty of Postgraduate Studies for Statistical Research, Cairo University, Cairo, Egypt
Email: osamah@bizegypt.com

How to cite this paper: Mansour, O.H., Raslan, A. and Ramadan, N. (2024) A Proposed Approach for Measuring Maturity Level of Software Delivery. *Journal of Software Engineering and Applications*, **17**, 228-245.
https://doi.org/10.4236/jsea.2024.175013

Received: April 3, 2024
Accepted: May 24, 2024
Published: May 27, 2024

Abstract

Software delivery is vital for modern organizations, driving innovation and competitiveness. Measuring an organization's maturity in software delivery is crucial for efficiency and quality. The Capability Maturity Model (CMM) framework provides a roadmap for improvement but assessing an organization's CMM Level is challenging. This paper offers a quantitative approach tailored to the CMM framework, using Goal-Question-Metric (GQM) frameworks for each key process area (KPA). These frameworks include metrics and questions to compute maturity scores effectively. The study also refines practices into questions for a thorough assessment. The result is an Analysis Matrix that calculates weighted scores and an overall maturity score. This approach helps organizations assess and enhance their software delivery processes systematically, aiming for improved practices and growth.

Keywords

CMM, CMMI, Software Delivery, Maturity, KPAs, GQM, Practices

1. Introduction

Software delivery is a critical aspect of modern businesses and organizations, serving as the conduit through which software products and services reach end-users. The efficiency and effectiveness of this process can significantly impact an organization's competitiveness, customer satisfaction, and overall success [1]. However, achieving a mature software delivery process is a complex endeavor, fraught with challenges that must be addressed to ensure optimal outcomes [2]. Numerous challenges plague the software delivery landscape, ranging from delays in project timelines and cost overruns to quality issues and stakeholder dissatisfaction. These challenges often stem from inadequate processes, unclear objectives, and a lack of standardized approaches to measure and improve software delivery maturity [3]. In the quest for a mature software delivery process, several fundamental questions emerge. Organizations seek to define and measure the maturity of their software delivery practices, understand the key metrics and indicators of maturity, leverage existing frameworks like the Capability Maturity Model "CMM" to enhance maturity levels, and identify actionable steps for process improvement [4].

This research aims to address these questions comprehensively. It begins by providing an overview of the challenges within software delivery, shedding light on the multifaceted issues faced by organizations in this domain. It then delves into the Capability Maturity Model "CMM" and its significance in this context [5]. Central to this research is the proposed method as a structured approach to measuring and enhancing software delivery maturity. The method consists of four layers: Refinement, Iteration, Development, and Knowledge. These layers work in tandem to guide organizations through the process of defining, measuring, and improving software delivery maturity.

2. Background

2.1. Capability Maturity Model

The Capability Maturity Model (CMM) is a framework initially developed to assess the maturity of the software development processes within an organization. It was created by the Software Engineering Institute (SEI) at Carnegie Mellon University in the late 1980s. The primary objective of CMM is to improve and optimize software development processes to enhance the quality of software products and the efficiency of development efforts [6].

CMM originated from the need to address the challenges of inconsistency and unpredictability in software development. Organizations were struggling to consistently produce high-quality software within time and budget constraints. The CMM framework aimed to provide a structured approach for organizations to assess and improve their software development practices systematically [7].

The history of CMM can be traced back to the late 1980s when the SEI was commissioned by the United States Department of Defense to address the quality and reliability issues in software development for defense projects. The initial version, CMM Level 1, focused on defining ad-hoc processes and aimed to bring stability to chaotic development environments.

In the realm of software development, the Capability Maturity Model (CMM) stands as a guiding framework, as described by Paulk *et al.* (1993) [6]. This model serves as a structured pathway for organizations to evaluate and enhance their software development processes. Comprising five distinct levels of maturity, the CMM offers progressive steps towards refined and well-defined processes [7].

Level 1, known as the Initial level, organizations often find themselves amidst ad hoc and unpredictable processes. Standard procedures are lacking, and reliance on individual efforts is prevalent.

Level 2, the Managed level, organizations begin to lay the groundwork with basic project management practices. Here, processes become more structured, with a focus on planning, tracking, and ensuring requirements are met.

Level 3, the Defined level, signifies a significant leap forward. Organizations at this stage have established well-defined and standardized processes. These processes are documented, understood, and consistently applied throughout the organization.

Level 4, the Quantitatively Managed level, organizations delve into the realm of measurement and control. Quantitative techniques are utilized to gain insights into processes, aiming for predictability and continual improvement.

Level 5, the Optimizing level, represents the pinnacle of maturity. Organizations here are in a constant state of refinement, always seeking ways to optimize and enhance their processes. They actively seek feedback, innovate, and base decisions on data to achieve maximum efficiency and quality [7].

In **Figure 1**, the more the framework is enforced on the process, the higher the maturity level of the organization as well as the outcomes resulting from the process. There are some fundamental concepts used in suing CMM described in **Figure 2** [8].

In **Figure 2**, the software process reflects many activities that are responsible for delivering the final product or software. There is an expected service or product that should result from those activities. The capability of the process refers to the expected product from the software process, while the process performance refers to the actual result or what really resulted from the software process.

Comparing the actual and expected results to determine the variance between them, or defining the extent to which they are equal, is the maturity of the process. So, the maturity level is usually high when the actual software delivered is close in nature and functionality to the expected result or product, while the maturity level is low when they are very different.

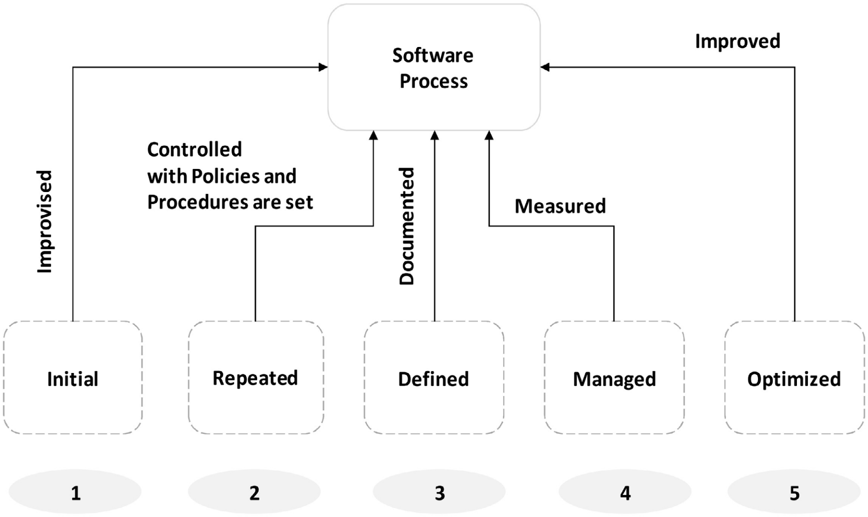

Figure 1. Capability maturity model fundamental concepts [4].

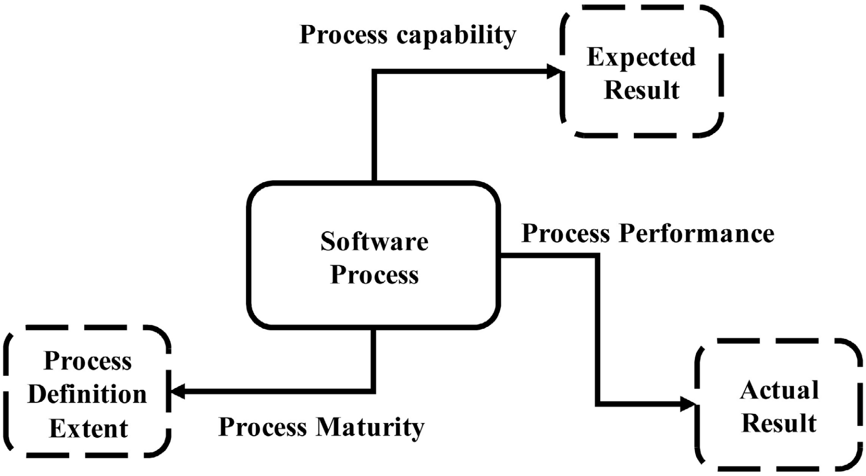

Figure 2. Capability maturity model fundamental concepts [5].

Maturity level is a number showing how well an organization is achieving the goals of the process, whether it is software or any other business process, as well as how well it is implementing the predefined practices set to achieve those goals.

2.2. Standard CMM Appraisal Method for Process Improvement

In the context of process improvement, the Standard CMM Appraisal Method "SCAMPI" emerges as a comprehensive and methodical approach. This method is deployed to evaluate and enhance the maturity of an organization's processes, specifically within the framework of the Capability Maturity Model Integration (CMMI) [9].

Moreover, SCAMPI serves as a structured framework, aligning an organization's processes with the defined goals and practices of the CMMI model.

The SCAMPI process unfolds through distinct stages, each integral to its efficacy. Beginning with meticulous planning, the appraisal team delineates the scope, objectives, and assessment criteria [10].

Subsequently, the appraisal phase ensues, involving the meticulous collection and analysis of evidence such as interviews, documentation, and observations. These efforts culminate in determining the maturity level of the organization's processes.

Following the assessment, the SCAMPI team generates a comprehensive report delineating the organization's process landscape. This report meticulously outlines strengths, weaknesses, and areas ripe for improvement. Serving as a strategic roadmap, it guides the organization towards implementing changes and enhancements, facilitating a progression towards heightened process maturity [9].

2.3. GQM

The Goal-Question-Metric (GQM) approach is a structured method used to define and measure goals in software engineering and other domains [1]. It establishes a hierarchy of goals, questions, and metrics, providing a systematic way to link an organization's goals to the metrics that can measure their achievement [10]. GQM emphasizes the importance of understanding what an organization wants to achieve, what questions need to be answered to achieve those goals, and what metrics can provide the necessary data [10].

The GQM method provides a framework for defining and measuring software quality.

The method is discussed as a practical guide for quality improvement in software development. It is designed to establish a set of goals and questions, which are then used to define metrics for measuring various aspects of the software development process. This includes measuring the quality of the product (software), the efficiency of the development process, and the utilization of resources [10].

In Figure 3, GQM can measure product, process, and resources as follows:

Product:

Goal: Improve the reliability of the software product.

Question: How reliable is the software product?

Metric: Defect Density (number of defects per line of code).

Process:

Goal: Enhance the efficiency of the development process.

Question: How long does it take to complete each development phase?

Metric: Cycle Time (average time to complete a phase).

Resources:

Goal: Optimize resource allocation for project tasks.

Question: How are resources allocated across different project tasks?

Metric: Resource Utilization Rate (percentage of time resources are utilized) [10].

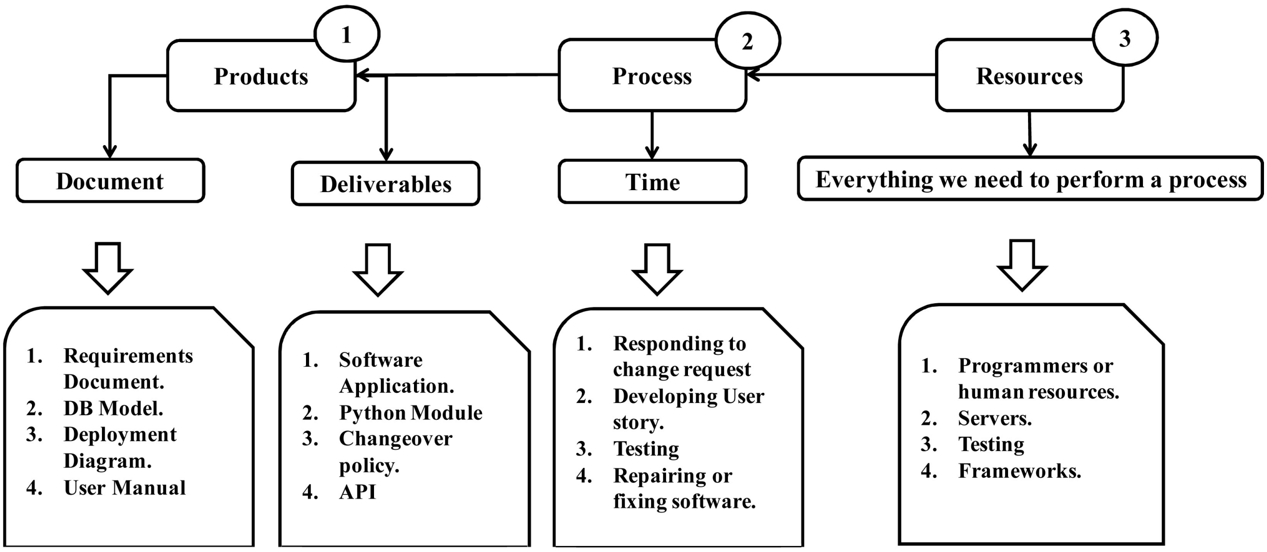

Figure 3. Different uses of GQM [10].

3. The Proposed Maturity Measurement Approach

The proposed approach is outlining a structured approach to measuring maturity in an organization, potentially using the Capability Maturity Model (CMM) framework along with its defined KPAs [11]. This approach aims to provide a systematic and structured method to assess and improve an organization's maturity across various KPAs. It combines elements of CMM, GQM, and a weighted scoring system to provide a comprehensive view of maturity [12].

3.1. Introducing the Proposed Approach

The proposed approach in **Figure 4** consists of four layers. The Refinement Layer is foundational, identifying, and refining goals with prioritization for effective strategies. The Iteration Layer then translates these goals into the GQM framework, creating metrics and computing maturity scores for tracking. Moving to the Development Layer, organization-wide maturity is calculated by weighting KPAs and combining their scores. The Knowledge Layer interprets data to craft a management report, offering insights for continuous improvement.

3.2. Refinement Layer

The "Refinement Layer" within the proposed approach serves as the initial step in the overall process, focusing on the meticulous enhancement of (KPAs). Through a systematic approach, this layer aims to clarify, prioritize, and refine the core processes that drive an organization's success.

This layer involves refining the (KPAs) at CMM Level 2.

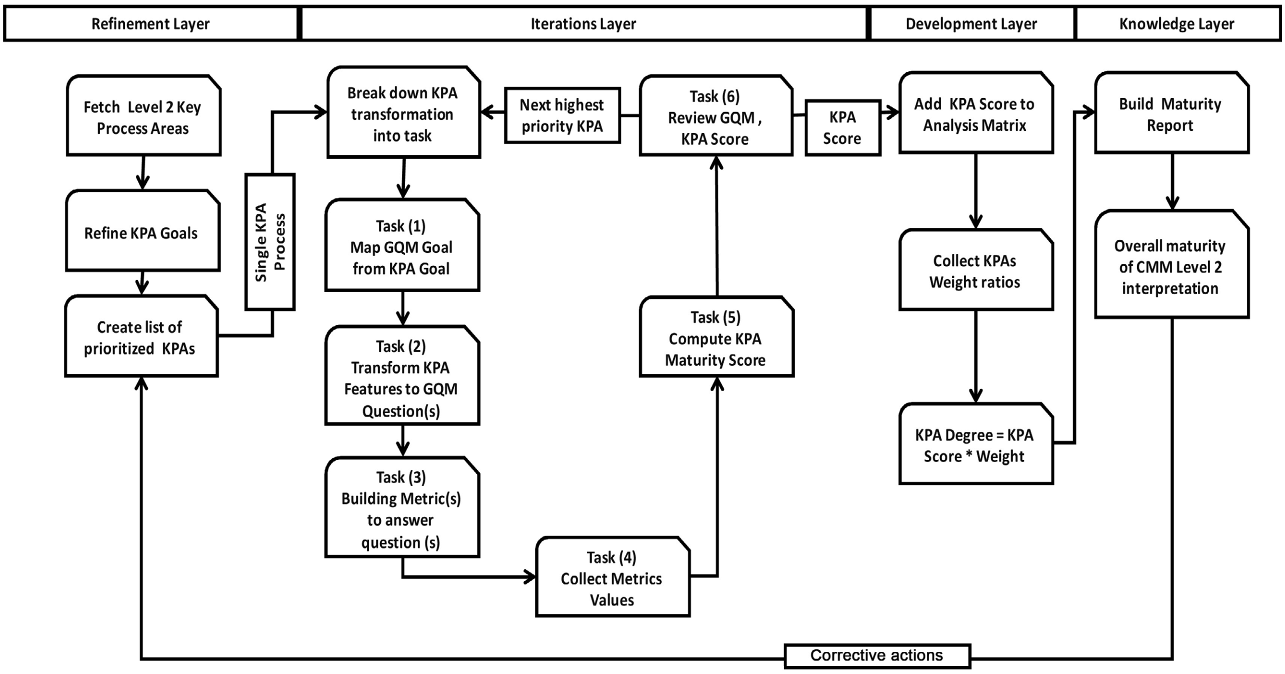
Refinement Layer
Iterations Layer
Development Layer
Knowledge Layer
Fetch Level 2 Key Process Areas
Refine KPA Goals
Create list of prioritized KPAs
Single KPA Process
Break down KPA transformation into task
Next highest priority KPA
Task (6) Review GQM, KPA Score
KPA Score
Add KPA Score to Analysis Matrix
Build Maturity Report
Task (1) Map GQM Goal from KPA Goal
Task (2) Transform KPA Features to GQM Question(s)
Task (3) Building Metric(s) to answer question (s)
Task (4) Collect Metrics Values
Task (5) Compute KPA Maturity Score
Collect KPAs Weight ratios
KPA Degree = KPA Score * Weight
Overall maturity of CMM Level 2 interpretation
Corrective actions

Figure 4. The proposed approach.

3.2.1. Fetch CMM Level 2 KPAs

In this step all KPAs of CMM Level 2 will be prepared for prioritization, according to [1]. The Level contains the KPAs show below.

Requirements Management: Involves identifying, documenting, tracking, and verifying requirements and changes throughout the project lifecycle.

Project Planning: Creating and maintaining a project plan that defines project activities, including scope, schedules, resources, and budget, to guide project execution.

Project Monitoring and Control: Tracks project performance against the plan, identifies issues, and takes corrective actions to keep the project on track, ensuring progress aligns with expectations.

Supplier Agreement Management: Involves selecting suppliers, establishing agreements, and ensuring acquired products meet specified requirements, managing the acquisition of products from suppliers.

Measurement and Analysis: Developing and sustaining a measurement capability to support management information needs by collecting and analyzing measurements for decision-making.

Process and Product Quality Assurance: Performing audits and reviews to provide objective insight into processes and work products, ensuring they meet specified quality standards.

Configuration Management: Establishes and maintains work product integrity using configuration identification, control, status accounting, and audits, managing configuration items and changes while ensuring integrity and traceability throughout the project.

3.2.2. Refine Goals of KPAs

Refining goals of collected KPAs is a process that is subject to clarify the goals associated with each KPA of the CMM framework. In **Table 1**, each KPA is listed along with its goal as defined [4].

3.2.3. KPAs Prioritization

Prioritize the KPAs based on their applicability, importance, or impact on the organization. Begin with the foundational KPAs that are essential for establishing a baseline of project management, KPAs that Address Risk and Quality Early, Project Monitoring and Control, Measurement, and analysis finally agreements with suppliers as shown below:

1) Project Planning (PP)—Establishes project baseline.

2) Configuration Management (CM)—Ensures work product integrity.

3) Requirements Management (RM)—Reduces misunderstandings.

4) Process and Product Quality Assurance (PPQA)—Ensures early quality.

5) Project Monitoring and Control (PMC)—Tracks project progress.

6) Measurement and Analysis (MA)—Provides data-driven insights.

7) Supplier Agreement Management (SAM)—Manages supplier relationships.

These are the KPAs we need to iterate through in the next layer to build the

Table 1. Goals of each key process area with CMM level 2 [8].

Key Process Area "KPA"	KPA Goal
Requirements Management (RM)	To manage requirements of products and product components and to identify inconsistencies.
Project Planning (PP)	To establish and maintain plans that defines project activities.
Project Monitoring and Control (PMC)	To provide an understanding of the project's progress for taking corrective actions.
Supplier Agreement Management (SAM)	To manage the acquisition of products from suppliers.
Measurement and Analysis (MA)	To develop and sustain a measurement capability to support management information needs.
Process and Product Quality Assurance (PPQA)	To provide staff and management with objective insight into processes and work products.
Configuration Management (CM)	To establish and maintain the integrity of work products using configuration management.

analysis matrix. Each GQM overall result "KPA Score" will be presented as row in the analysis matrix [1].

3.3. Iteration Layer

This layer focuses on translating the goals of each KPA into specific objectives, converting "Practices" into questions, and defining quantitative metrics to answer these questions as part of GQM approach. In a real-world scenario or when implementing this approach, we should follow the steps outlined from 3.3.1 to 3.3.6 for each KPA. However, for simplicity, we will conduct a measurement the first KPA "Project Planning" to illustrate the approach. The same steps can be followed in a similar manner but within a different context or in other words on the reset of KPAs.

3.3.1. Mapping KPA Goal into GQM Goal

Translate the goals of each KPA into corresponding goals in the GQM framework.

As we have mentioned earlier in the background part GQM goal should contains essential properties, thus we need to reproduce each KPA goal in proper GQM goal.

KPA Goal: The goal of Project Planning (PP) is to establish and maintain plans that define project activities. This includes outlining the scope, activities, schedules, resources, and budget for the project.

GQM Goal: Improve project planning accuracy and efficiency.

Explanation: The GQM goal for Project Planning (PP) is to enhance the accuracy and efficiency of project planning processes. This involves ensuring that project plans are comprehensive, realistic, and well-structured to guide project execution effectively.

GQM Goal Information:

- *Purpose*: The reason for creating the GQM Goal, which is to assess the maturity level of the KPA.
- *Issue*: The problem or challenge being addressed, which is the challenge of defining KPA maturity.
- *Object*: The entity or subject of the measurement, which is the Key Process Area.
- *Viewpoint*: The perspective from which the assessment is made, in this case, the viewpoint of the Project Manager [13].

3.3.2. Transform KPA Features into GQM Questions

Define specific questions (related to the GQM goals) that will be used to assess the achievement of KPA goals. This step is essential because KPA features will be transformed into questions that can have measurable metrics [10]. For instance the feature (Establish project scope) can be transformed into the question (Are project plans aligned with the defined project scope?).

In Table 2, GQM questions are designed to assess the key features of Project Planning (PP) and ensure that they contribute to the overall goal of improving project planning accuracy and efficiency. Each question targets a specific aspect of project planning that is essential for successful project execution.

Table 2. KPA Features transformed into GQM Questions [8].

KPA Feature	GQM Question
Establish project scope	Are project plans aligned with the defined project scope?
Define project activities and tasks	Are project tasks and activities clearly defined in the plan?
Create project schedules	Is the project schedule realistic and achievable?
Allocate resources appropriately	Are resources allocated optimally based on project requirements?
Define project budget	Is the project budget accurately reflected in the plan?
Identify project risks and mitigation plans	Are project risks identified and mitigation plans in place?
Develop communication and reporting processes	Are communication channels and reporting processes well-defined?
Establish project milestones	Are project milestones clearly defined and tracked in the plan?

3.3.3. Build GQM Metrics

Specific metrics should be created to answer all GQM Questions as shown in **Table 3** [11].

Table 3. GQM Questions and related metrics for project planning KPA [10].

GQM Question	GQM Metrics
Are project plans aligned with the defined project scope?	- Percentage of project scope defined in the plan.
Are project tasks and activities clearly defined in the plan?	- Number of undefined or ambiguous tasks.
Is the project schedule realistic and achievable?	- Deviation from Planned Timeline (in days or percentage).
Are resources allocated optimally based on project requirements?	- Accuracy of Resource Allocation (percentage of tasks with correct resource allocation).
Is the project budget accurately reflected in the plan?	- Percentage of project budget allocated to tasks.
Are project risks identified and mitigation plans in place?	- Number of identified risks.
Are communication channels and reporting processes well-defined?	- Number of communication channels established.
Are project milestones clearly defined and tracked in the plan?	- Percentage of completed activities on schedule.

3.3.4. Set Metrics Values (Collect Data)

Collected Data should reflect the actual organization status about each metric that can be achieved through onsite visits to the organization, interviewing, observation, and documentation audit or reviewing relevant documents such as reports, records, policies, and procedures offers a comprehensive view of how metrics are being recorded and utilized. It helps in cross-referencing data obtained from other methods and ensures consistency in information. By employing these diverse data collection methods, a more holistic and accurate understanding of the organization's status in relation to each metric is achieved. This detailed approach enables informed decision-making and the identification of areas for improvement within the organization's processes.

3.3.5. Compute KPA Maturity Score

After collecting metrics answers we can calculate the maturity score for the KPA.

Below is the full GQM along with KPA features, Questions, Metrics, unit of each metric, metric value collected and scale from 0 to 5 where 0 is not achieved at all and 5 is fully achieved [12].

In **Table 4**, the components of the KPA are then analyzed with their associated questions, metrics, units, values, and points of practice or feature [10]. For example the Feature (Practice) Define Project Scope has the following details:

- Question: Is the project scope clearly defined?

Table 4. GQM for project planning KPA [10].

KPA: Project Planning.
KPA Goal: To establish and maintain plans that defines project activities. This includes outlining the scope, activities, schedules, resources, and budget for the project.
GQM Goal: Improve project planning accuracy and efficiency.

Goal	**Purpose**: The reason for creating the GQM, assess maturity level of the KPA. **Issue**: The problem being addressed, defining KPA maturity. **Object**: The entity or subject of the measurement (Key Process Area). **Viewpoint**: Project Manager.

KPA Feature	Question	Metric	U	V	P	Level
Define Project Scope	Is the project scope clearly defined?	Clarity of Project Scope	%	30	2	Repeated
Establish Project Schedule	Is there a detailed project schedule?	Completeness of Project Schedule	%	10	1	Initial
Define Work Breakdown Structure	Is there a work breakdown structure (WBS)?	WBS Completion	Num	2	2	Repeated
Estimate Effort and Resources	Are effort and resources estimated?	Accuracy of Effort and Resource Estimates	Num	1	1	Initial
Identify Dependencies	Are project dependencies identified?	Completeness of Dependency Identification	Num	2	2	Repeated
Establish Risk Management Plan	Is there a risk management plan?	Existence of Risk Management Plan	Num	4	4	Managed
Allocate Responsibilities	Are responsibilities clearly assigned?	Clarity of Responsibility Allocation	Num	3	3	Defined
Define Quality Plan	Is there a quality plan for the project?	Existence of Quality Plan	Num	5	5	Optimized
KPA Score					**2.5**	**Repeated**

- Metric: Clarity of Project Scope.
- Unit (U): Percentage (%) of clear and well organized and documented projects.
- Value (V): 30%.
- Point (P): Scale from 0 to 5 (2).
- Level: Repeated, the commentary column can present directly which practice needs improvement.

The final KPA Score, in this case 2.5, represents the overall maturity level of the Project Planning KPA, derived from the assessment of its individual features. This score falls between the categories of "Repeated" and "Defined", indicating the current maturity level of the KPA in terms of accuracy and efficiency. Organizations usually fall into this rank, has achieved basic project management processes. These processes are defined, documented, and followed consistently across projects [8]. This value will be appended to the analysis matrix in the next step for further computations so that we can find out the company accumulated score [14].

3.3.6. Append KPA Maturity Score to the Analysis Matrix

Add the computed maturity scores to an analysis matrix, which likely tracks all KPAs and their respective scores [14]. As result of iterating through key process area of CMM level 2, the analysis matrix will be completed at the end. The targeted KPAs are Project Planning, configuration Management, Requirement Management, Product and Process Quality Assurance, Project Control Management, Supplier Agreement Management Finally Measurement and Analysis.

3.3.7. Assign Weight of KPAs

Weighted scores are typically calculated based on the relative importance or significance of each Key Process Area (KPA) within the context of the assessment or project. There isn't a universal standard for assigning weights, as it largely depends on the specific criteria, objectives, and priorities of the project or assessment.

3.4. Knowledge Layer

The Knowledge Layer is where the raw data transforms into actionable knowledge. It involves interpreting the analysis matrix, deriving business insights, visualizing data for clarity, organizing through classification, and utilizing GQMs for strategic alignment. This layer bridges the gap between data and decision-making, empowering businesses with informed choices and strategic directions.

Create Maturity Report

The Maturity Report is not just a numerical representation; it's a roadmap to organizational excellence. It leverages the analysis matrix, GQM insights, and CMM framework to provide a comprehensive view of where the organization stands and where it can go.

Interpret the data in the analysis matrix to understand the strengths and weaknesses of the organization's maturity in each KPA [14]. In **Table 5** Project Planning the Weight is 0.15 refer to the applicability rate set by the organization for project planning KPA, Score is 4 this score is received from GQM of project planning, Weighted Score is the product of Weight (0.15) * score (4) = 0.6. Now let us sum up all weighted scores to compute the overall maturity score of the organization against the requirements of CMM framework level 2. So, the total CMM Level 2 Maturity Score is 2.17.

Ultimately, a comprehensive analysis of KPA maturity provides a roadmap for strategic growth and optimization, guiding organizations towards enhanced processes and better outcomes [14]. For instance understanding the maturity of KPA is paramount to illuminate the strengths and weaknesses of an organization's maturity across various KPAs. The chart below is extracted from the data within the analysis matrix; it is presenting how each KPA contribution in the overall organization's maturity towards achieving improved processes and matches the requirements of CMM framework.

Table 5. CMM Level 2 KPAs analysis matrix [14].

ID	KPA	Weight	Score	Weighted score
1	Project Planning (PP)	0.15	4	0.6
2	Configuration Management	0.1	3.6	0.3625
3	Requirement Management	0.12	1.7	0.204
4	Product and Process Quality Assurance	0.08	3.7	0.296
5	Project Control Management	0.13	1.7	0.221
6	Measurement And Analysis	0.11	0.5	0.055
7	Supplier Agreement Management	0.09	4.8	0.432
	Total CMM Level 2 Maturity Score			**2.17**

Figure 5 provides a visual snapshot of the CMM Level 2 Maturity Scores for each KPA, allowing stakeholders to quickly identify areas of strength and potential improvement within the software delivery process. Supplier Agreement Management has the highest Weighted Score of 0.432, indicating it is a strong area within the CMM Level 2 assessment. Project Planning (PP) follows closely with a Weighted Score of 0.6, showing good maturity in this area. Configuration Management and Product and Process QA are also relatively strong with scores of 0.3625 and 0.296, respectively. Requirement Management and Project Control Management have moderate scores, indicating there may be some room for improvement in these areas. Measurement And Analysis has the lowest Weighted Score at 0.055, indicating it may be a weaker area that needs attention. Frequency table may be developed to discover the state of practices within the organization against the different CMM levels as show below:

In Figure 6 we graphically resent the data in Table 6. The ratios provided in the table, which indicate the distribution of practices within each Capability Maturity Model (CMM) level, hold significance for businesses. They offer insights into how resources and efforts are allocated at different stages of process maturity. For instance, the ratio of 24% for Initial indicates that out of the total practices identified, 24% are dedicated to ad hoc processes without much formalization or consistency. This suggests that at this stage, a significant portion of the organization's efforts may be focused on reacting to immediate needs rather than following structured, standardized processes. Moving to the Defined level with an 11% ratio, businesses allocate fewer resources to documentation and standardization compared to the Initial stage. This suggests a shift towards more formalized processes with less emphasis on the sheer volume of practices and more on ensuring they are well-documented and standardized. The Managed level, with a ratio of 19.5%, indicates an increased focus on collecting process metrics to manage and control operations. Businesses at this stage are investing resources in data-driven decision-making and process control, which can lead to improved efficiency and quality. At the Repeated level, the ratio jumps to 26%,

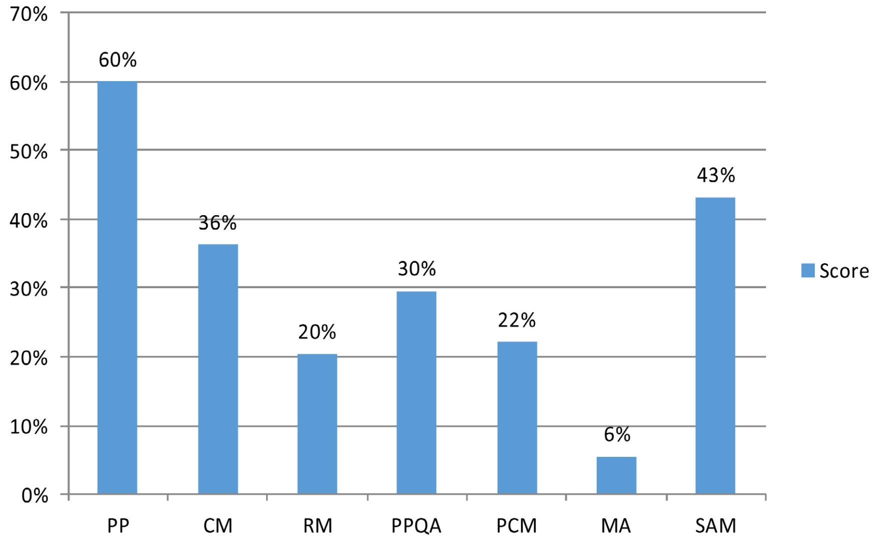

Figure 5. Organization's maturity to deliver software.

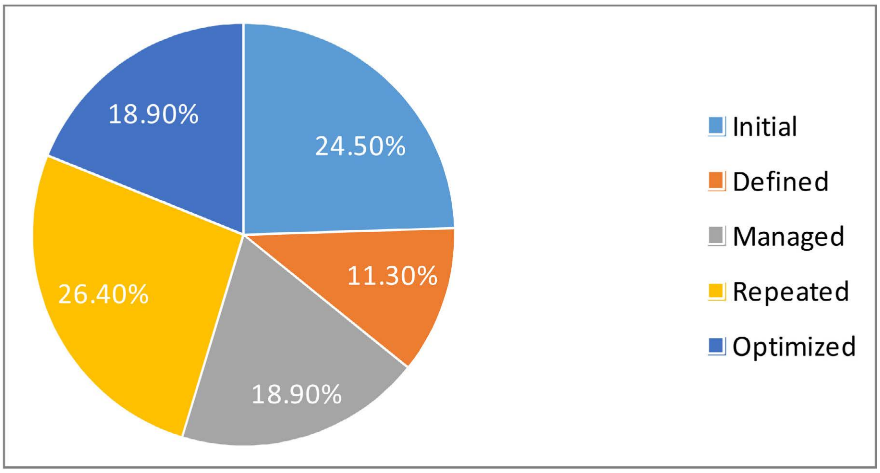

Figure 6. Distribution of the organization's practices among maturity levels.

Table 6. Organization's rates in among CMM Levels [13].

CMM Level	Practices	Ratio
Initial: Ad hoc processes without much formalization or consistency.	13	24.5%
Defined: Processes are documented and standardized throughout the organization.	6	11.3%
Managed: Process metrics are collected and used to manage and control processes.	10	18.9%
Repeated: Processes are consistently executed with a focus on quality and efficiency.	14	26.4%
Optimized: Continuous process improvement is ingrained in the organization's culture.	10	18.9%

showing a significant portion of practices dedicated to consistent execution with a focus on quality and efficiency. This suggests that businesses are prioritizing the refinement and optimization of their processes to achieve reliable results. Finally, the 19.5% ratio for the Optimized level indicates that despite the lower number of practices compared to Repeated, businesses are dedicating a substantial portion of their efforts to continuous process improvement ingrained in the organizational culture. This signifies a shift towards innovation, adaptability, and ongoing optimization, leading to sustained success and competitive advantage.

4. Limitations

The method outlined in the article appears comprehensive and structured, but like any approach, it may have limitations. Below are some potential issues and suggestions on how to address them.

4.1. Subjectivity in Prioritization

The prioritization of KPAs may involve subjective judgment, which can vary among evaluation team. To mitigate this, ensure that the prioritization process involves input from diverse stakeholders representing different perspectives within the organization. Additionally, consider incorporating objective criteria, such as impact on business objectives or alignment with strategic goals, to guide prioritization decisions [15].

4.2. Reliance on Self-Assessment

The method relies on self-assessment, which may introduce biases or inaccuracies in the evaluation process. To avoid this, consider supplementing self-assessment with external validation mechanisms, such as independent audits or third-party reviews, to provide a more objective perspective on the organization's maturity level.

4.3. Limited Scope of KPAs

The method primarily focuses on KPAs defined within the CMM framework, [16], which may not fully capture all relevant aspects of organizational maturity, particularly in rapidly evolving domains such as software development. To address this, consider complementing the CMM framework with additional maturity models or frameworks that address specific areas of concern or emerging trends in the industry [17].

4.4. Difficulty in Sustaining Improvement

Sustaining long-term improvement efforts may be challenging. It is possible to establish mechanisms for continuous monitoring and review of maturity levels, set realistic improvement targets, and integrate maturity assessment into regular business processes to ensure ongoing focus and commitment to improvement

[18].

4.5. Difficulty in Assigning Weight of KPAs in Analysis Matrix

A notable limitation of the evaluation process lies in the challenge of accurately defining the weight or importance rate assigned to each Key Process Area (KPA) by the evaluation team and stakeholders. To address this issue, organizations can enhance the maturity assessment process through stakeholder engagement, data-driven decision-making, and transparency in documentation, in addition ensure that the weighting criteria accurately reflect organizational priorities and facilitate meaningful insights for continuous improvement [15].

5. Effect of Applying the Approach

Let us delve deeper into the significant effect of applying the proposed method across multiple organizations to create a competitor evaluation matrix, which showcases the position of each firm in terms of software delivery maturity:

1) **Standardized Assessment Framework:** The method offers a structured and standardized approach to assessing software delivery maturity across various organizations.

2) **Comparative Analysis:** By implementing this method across multiple organizations within the same industry or market segment, it becomes possible to conduct a comparative analysis of their software delivery maturity levels.

3) **Competitor Evaluation Matrix Creation:** The collected data can then be used to construct a competitor evaluation matrix, which visually represents the position of each firm in terms of software delivery maturity [15].

4) **Identification of Strengths and Weaknesses:** By examining the maturity scores within the competitor evaluation matrix, organizations can identify their strengths and weaknesses compared to competitors across different KPAs. This insight enables them to focus on leveraging their strengths and addressing areas of improvement to enhance overall software delivery maturity [15].

5) **Benchmarking:** The competitor evaluation matrix facilitates benchmarking exercises, allowing organizations to compare their performance against industry standards and leading competitors.

6) **Strategic Decision-Making:** Armed with insights from the competitor evaluation matrix, organizations can make more informed strategic decisions regarding resource allocation, process optimization, and technology investments [15].

6. Conclusions

In closing, the main findings of this approach revolve around understanding the organization’s current maturity level, identifying areas for improvement, and providing actionable insights for continuous growth and optimization. The paper has introduced a maturity measurement method for assessing the software delivery process within organizations, and quantitatively defined its maturity level as well as the different maturity across KPAs.

The significance of this approach lies in its ability to provide organizations with a structured, objective, and data-driven method for assessing and improving organizational maturity. By leveraging established frameworks and methodologies, organizations can optimize their processes, drive efficiency, and achieve long-term success. It provides researchers and practitioners with a structured framework for assessing, analyzing, and enhancing organizational capabilities of software delivery.

The proposed approach integrates established framework CMM and GQM approach. This integration contributes to the advancement of organizational maturity theory by providing a structured and systematic method for assessing and improving maturity of software delivery. It highlights the importance of aligning organizational practices with industry standards and best practices for better governance. Policymakers can encourage the adoption of recognized frameworks like CMM to promote consistency and quality in organizational processes.

However, some limitations for instance subjectivity in prioritizing KPAs, self-assessment reliance may introduce biases, focusing primarily on CMM-defined KPAs may miss crucial aspects finally assigning weights to KPAs in the analysis matrix is not easy.

It is evident that implementing this approach on rest of CMM levels that are not involved in this paper would also be an interesting area for future research. In addition, assessing software delivery maturity across multiple organizations by implementing the approach to facilitate the creation of a competitor evaluation matrix in context of software delivery or even in any other context is a possible future research area and holds promise for advancing organizational excellence in software delivery and beyond.

Acknowledgements

R. B. G. thanks Dr. Atef Raslan and Prof. Dr. Nagy Ramadan for their invaluable guidance, insightful feedback, and continuous encouragement throughout this research. Their expertise and mentorship have been instrumental in shaping this work.

Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

References

[1] Lu, K. (2021) Modern Software Delivery Maturity Matrix. Digital Service Start-Ups. https://medium.com/digital-service-start-ups/modern-software-delivery-maturity-matrix-2be5e840f0c4

[2] CMMI Product Team (2023) CMMI V2.0: Advanced Topics in Process Improvement. Carnegie Mellon University Software Engineering Institute.

[3] Khraiwesh, M. (2017) Configuration Management Measures in CMMI. *International Journal of Applied Engineering Research*, **12**, 7546-7557.

[4] Stelzer, D. (2005) Software Process Improvement with CMM. Springer Science & Business Media, Berlin.

[5] Paulk, M.C., Weber, C.V., Garcia, S., Chrissis, M.B. and Bush, M. (1993) Key Practices of the Capability Maturity Model, Version 1.1. Software Engineering Institute, CMU/SEI-93-TR-25. https://insights.sei.cmu.edu/documents/1093/1993_005_001_16214.pdf

[6] Watts, S. (2020) CMMI: An Introduction to Capability Maturity Model Integration. BMC. https://www.bmc.com/blogs/cmmi-capability-maturity-model-integration/

[7] Weber, C.V., Paulk, M.C., Wise, C.J. and Withey, J.V. (1991) Key Practices of the Capability Maturity Model. Software Engineering Institute, CMU/SEI-91-TR-25, ADA240604.

[8] SCAMPI Upgrade Team (2011) Standard CMMI Appraisal Method for Process Improvement (SCAMPISM). Software Engineering Process Management.

[9] Braungarten, R. (2007) The SMPI Model: A Stepwise Process Model to Facilitate Software Measurement Process Improvement along the Measurement Paradigms. Doctor's Thesis, Otto von Guericke University, Magdeburg. https://d-nb.info/985144955/34

[10] Basili, V.R., *et al.* (1994) Goal Question Metric Paradigm. *Encyclopedia of Software Engineering*, **2**, 2. https://www.cs.umd.edu/users/mvz/handouts/gqm.pdf

[11] Yahya, F., Walters, R.J. and Wills, G.B. (2017) Using Goal-Question-Metric (GQM) Approach to Assess Security in Cloud Storage. In Chang, V., *et al.* (Eds.), *Enterprise Security*, Springer, Berlin, 223-240.

[12] Van Veenendaal, E., and van der Zwan, M. (2001) GQM Based Inspection. https://www.researchgate.net/publication/2537262_GQM_based_Inspection

[13] Yamfashije, J. (2017) Capability Maturity Model Integration. https://www.researchgate.net/publication/327557963_Capability_Maturity_Model_Integration

[14] Irigoyen, A. and Montoni, M. (2020) Capability Counts Series: Reaching CMMI Engineering Practices with DevOps. https://cmmiinstitute.com/news/blog/capability-counts-series-reaching-cmmi-engineering

[15] Dogaru, M. and Chira, R. (2013) Analysis of Internal Environment. *Social Economic Debates*, **2**, 10-15. https://www.economic-debates.ro/nr2-2013/art2-dogaru.pdf

[16] Frisina, L. (2022) Software Delivery Metrics You Cannot Afford to Miss! DEVOPS. https://www.adaptavist.com/blog/delivery-metrics-you-cant-miss

[17] Klein, J., Konrad, M.D. and Stoddard, R.W. (2020) Beyond Process Improvement: Leveraging CMMI for Business Excellence. Springer, Berlin.

[18] Giray, G. (2021) A Software Engineering Perspective on Engineering Machine Learning Systems: State of the Art and Challenges. https://www.researchgate.net/publication/346405023_A_Software_Engineering_Perspec-tive_on_Engineering_Machine_Learning_Systems_State_of_the_Art_and_Challenges