

Providing Guidance to Software Practitioners

A Framework for Creating Key Performance Indicators

Iffat Fatima¹, Markus Funke², and Patricia Lago³, Vrije Universiteit Amsterdam

// We propose a framework in the form of a model, an associated template, and a checklist to provide guidance to practitioners for creating sound key performance indicators, examining their creation through four case studies in the context of software systems in two organizations. //



©SHUTTERSTOCK.COM/THAPANA_STUDIO

YOU CANNOT IMPROVE what you cannot measure. This philosophy by Peter Drucker¹ resonates with organizations that actively seek to drive continuous improvement in performance by collecting pools of data. However, measuring for the sake of measuring is a common pitfall observed in various organizations. Measurement should focus on capturing information that is specifically needed and adds value to organizational goals, rather than measuring everything indiscriminately. Therefore, it is crucial to establish a link between goals and measurements by using a structured mechanism.

Kaplan and Norton² introduced the concept of key performance indicators (KPIs) in the context of building balanced score cards (BSCs) to translate strategy into action. However, they did not present a concrete definition of KPIs. Parmenter³ defines a KPI as “an indicator that focuses on the aspects of organizational performance that are the most critical for the current and future success of the organization.” In the digitized organizational landscape, monitoring KPIs for software systems has become increasingly important. Hence, Parmenter’s definition should be reimagined to focus on the “...aspects of software performance...” instead.

Due to the fluid nature of business goals, it becomes complex to link these goals to concrete software measurements. KPIs can provide potential support to fill this gap. To provide this as a practical guideline, we present a framework comprised of a KPI template for crafting KPIs and a practical KPI checklist for evaluating their soundness, based on our KPI model. The framework is grounded in the relevant literature on KPI

definitions and is applied to four software-related industrial case studies.

We aim to use the business needs of the organizations to motivate focused measurement of data. We provide a KPI framework to visualize the KPI creation process and aid practitioners in using it in conjunction with their own measurement and strategy frameworks. The template can be used to 1) create new KPIs from scratch, and 2) decompose the existing KPIs to see their mapping to goals and source of measurement information, and identify any gaps.

Our investigation examines the organization's current practices and limitations related to KPI creation and monitoring. The case studies evaluate the effectiveness of our model and checklist by mapping them to real-world KPIs. On the other hand, the checklist acts as a validation mechanism for the sound definition of the organization's KPI via the template. Finally, we present practical insights as a companion for day-to-day work with KPIs in a software-related context.

Crafting a KPI Creation Framework

We created a theoretical KPI model in UML notation [Figure 1(a)]. The KPI model results from both fundamental building blocks from the literature and current limitations of existing measurement theory. In "Overview and, we first discuss the prominent literature, then we outline how we employed semi-structured interviews to evaluate and improve our proposed KPI framework, to finally frame and examine the fundamental framework building blocks.

Model as Backbone

"All models are wrong, but some are useful" is a concept often attributed

to George E. P. Box.⁴ Despite not fully capturing system complexity, models can identify patterns, serving as tools for analysis and decision-making.⁵ In line with this perspective, we created a KPI model [Figure 1(a)] to represent the relationship between the conceptual building blocks and their usage, supporting practical and actionable decision-making.

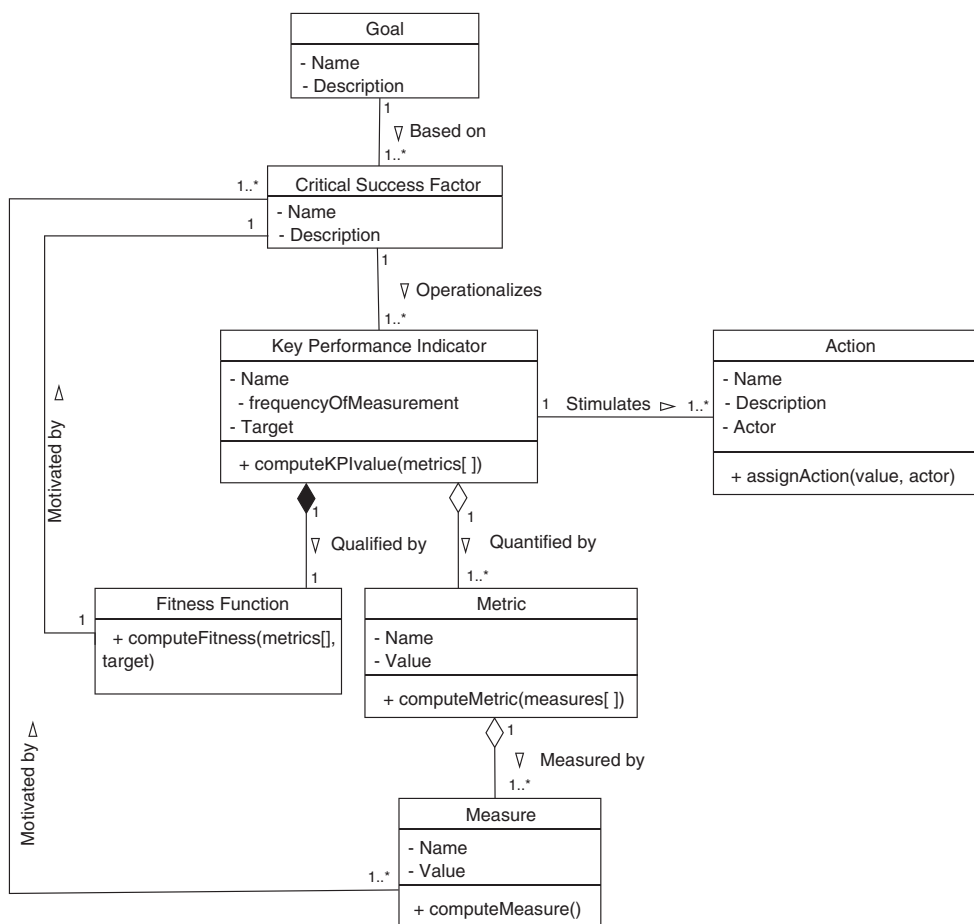
Template as Crafting Tool

From the KPI model we derived a KPI template [see Figure 1(b)] to increase the model's utility for practitioners. Using this template, we conducted case studies to brainstorm with practitioners to translate their mental model of KPIs into our tangible KPI model.

Checklist as Self-Assessment

To facilitate the correct usage of our KPI template, we supplemented it with a KPI checklist based on Parmenter's work.³ The checklist [see Figure 1(c)] ensures the soundness and correct usage of different elements of the KPI model as a validation mechanism. While we acknowledge the significance of the SMART⁶ (specific, measurable, assignable, realistic, time-related) evaluation method in defining focused goals, we must also recognize its limitations when it comes to defining sound KPIs. The generic nature of SMART criteria hinders their operationalization. For instance, the "specific" criterion requires a clearly stated objective; the interpretation of clarity remains subjective and varies across different contexts. Instead, we favor the use of objective checklist items that are tailored for their usage in practice and directly align with our model. The checklist items are described as follows.

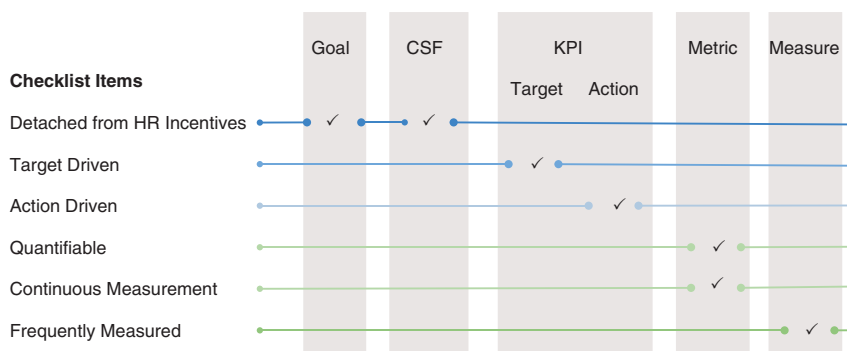
- *Detached from human resource (HR) incentives:* Parmenter³ emphasizes that employee rewards should not be associated with organizational performance, as it can lead to the creation of loopholes in the reward system. Continuing with the example from the building blocks, the goal "ensure end-users with access to IT services without disruption" should not be hardwired to set employee bonuses. Rather the focus should be on service availability to end users. Similarly, for critical success factor (CSF) "solving IT incidents within agreed timelines," a team should not be rewarded based on the number of incidents resolved, as this can lead to misuse of the KPI framework.
- *Target driven:* Each KPI must have an associated quantifiable target value based on organizational goals. A KPI without a quantifiable target value becomes delinked from organizational goals and strategy.
- *Action driven:* KPIs must be action-driven: i.e., if a KPI is not meeting its target, the organization should know what action(s) need to be initiated and who is responsible for those actions.
- *Quantifiable:* All KPIs must be quantifiable. If a metric cannot be quantified, it is not relevant for KPI measurement.
- *Continuous measurement:* A KPI only measures current performance. Based on Parmenter's³ view, a KPI has a continuous value and is different from the key results that may be based on past data. A KPI is like a speedometer on a car's dashboard that tells you how quickly you will reach your target. This can only



(a)

<KPI Name>					
Goal	Critical Success Factor	Key Performance Indicator		Metric	Measure
		Target	Action		

(b)



(c)

FIGURE 1. KPI framework. (a) KPI model, (b) KPI template, and (c) KPI checklist. HR: human resources.



OVERVIEW AND METHODOLOGY

RELATED WORK ON SOFTWARE MEASUREMENT

The software measurement literature is rich in measurement theory and definitions.^{S1,S2} However, these measurements should be intentional, goal-focused, and target-driven.

Basili et al.^{S3} present a goal question metric approach to identify goal-focused metrics. With the goal question metric, the system performance is represented at a granular level with scattered data that can be aggregated as a goal-focused, target-driven key performance indicator (KPI) by using our framework.

Kaplan and Norton² present the idea of using a balanced score card (BSC) to observe a holistic view of the system performance across predefined chosen perspectives (e.g., financial, customer, internal business processes, learning, and growth, etc.). Parmenter³ uses critical success factors (CSFs) to identify the measures for BSCs. We use a similar approach to define KPIs per CSF without linking them to perspectives. It is up to the organization to link these KPIs to a broader strategy framework like BSCs, strategic maps, performance prism, etc.

With our framework, we provide a blank template for practitioners to work with, eliminating any bias created by predefined perspectives. Practitioners can use this template to fill in the information about strategic elements of our KPI framework in conjunction with the internally used strategy frameworks. We do not intend for our KPI framework to replace these established tools. Instead, our goal is to assist organizations in developing KPIs with seamless integration, using our framework as a supportive tool that complements, rather than disrupts, their current workflow.

Quite some variation is observed in measurement taxonomy (e.g., metrics,^{S3} composite metrics,^{S1} derived measures^{S2}). This leads to collecting nonuniform data and creating noncomparable indicators. The measurement information model from International Organization for Standardization/International Electrotechnical Commission (ISO/IEC) 15939^{S4} provides a model that aggregates the measurement information via base and derived measures to indicators that represent the information needed to manage goals, risks, and problems of the relevant system. With our framework, we break down these information needs from the measurement information model^{S4} into

our building blocks (like measures and indicators) and concepts (like goals and CSF) from strategy frameworks that motivate the measurement process.

Using the strengths of strategic management tools and measurement theory, we aim to create a KPI template that practitioners can intuitively use. Our framework aims to provide a terminological separation of concerns facilitating a systematic approach to link business needs to measurable properties of software for performance monitoring. We link the KPIs to strategic targets and corrective actions, which makes the framework practical and action-driven.

A structured communication of organizational performance has become imperative with upcoming regulations about data transparency of digital products. For instance, with the emergence of the European Union Corporate Sustainability Reporting Directive, organizations are now required to report CO₂ emissions and are obligated to meet targets by 2025 for which target-based KPIs are much needed.

STUDY METHODOLOGY

Study Design

We used the literature to build the theoretical foundation of our KPI model in the form of a UML diagram [Figure 1(a)]. These building blocks helped us to construct the first draft of the KPI template, which was used as input for the semi-structured interviews having two goals: 1) use

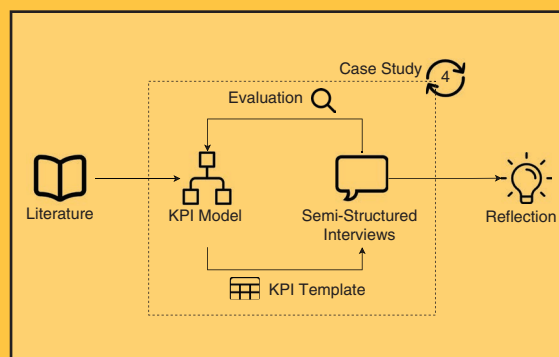


FIGURE S1. Study methodology.

(Continued)

OVERVIEW AND METHODOLOGY (CONTINUED)

practitioners' feedback and their different real-world KPIs to evaluate and improve our framework; and 2) support the practitioners in evaluating their KPIs and filling any potential inconsistencies or gaps uncovered by using the KPI framework. We carried out this evaluation and improvement loop by conducting four case studies. For each case study, we used semi-structured interviews to collect data to populate, evaluate, and reflect on the KPI template (Figure S1).

To assess our framework's applicability, we investigated KPI creation in a financial institute and an aviation organization, focusing on their software systems. In the financial institute, we engaged with practitioners directly responsible for KPI creation and monitoring. In the aviation organization, in turn, we engaged with the sustainability officers and the project team responsible for monitoring. We identified the missing (red), newly discovered (grey), and documented (blue) elements in the filled-out templates as shown in Table 1. For defining the KPIs, we used a hybrid approach (mix of top-down and bottom-up) in the first three case studies, while in the last case study, we used a bottom-up approach. We concluded this exercise by reflecting on the lessons learned.

Model Evaluation

We conducted semi-structured interviews for each 'improvement' iteration (protocol accessible at <https://bit.ly/4bNhGqq>). We held four sessions, each focusing on a specific KPI chosen by the practitioners before the session. Two sessions had two participants each, while the other two had one participant each. Interviews were transcribed using otter.ai (see <https://otter.ai/>) and revised by the first and second authors resulting in an edited transcription. As a qualitative research method, we applied deductive coding together with structural coding based on the interview protocol to analyze the transcriptions. During the interviews, we shared the KPI template with participants to fill it out collaboratively. This method provided in-depth insights into the KPI process for each case. Moreover, this step demonstrated and evaluated how our KPI framework performs in practice.

BUILDING BLOCKS AS A FOUNDATION

We present the definition of the key terms that act as building blocks (Figure S2), each illustrated with a concrete example.

- **Goal:** "Goal is a long-term outcome that your teams use to help define strategies."^a Example: Ensure end-users with access to IT services without disruption.
- **CSF:** "Critical success factors are the limited number of areas in which results, if satisfactory, will ensure successful competitive performance for the organization."⁵ CSFs of the organization are based on the goals. Example: Solving IT incidents within agreed timelines.
- **KPI:** "An indicator that focuses on the aspects of software performance that are the most critical for the current and future success of the organization."³ A CSF should be operationalized by one or more KPIs. Example: Percentage of out-of-time incidents. A KPI has a fitness function that checks the conformance of the KPI value to the KPI target. A KPI is qualified if the value meets the target. Example: Fitness function checks with every new incident report if the KPI value fell below target. If so, action is triggered. If a fitness function is not satisfied with a certain KPI target, the KPI stimulates action(s) in the organization. An actor is assigned to

^a<https://digitalleadership.com/blog/business-goals/>

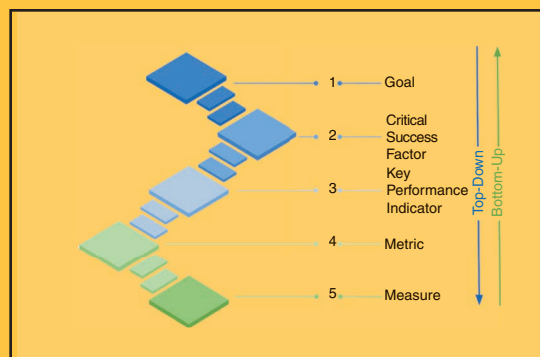


FIGURE S2. Building blocks of KPI model.

(Continued)



OVERVIEW AND METHODOLOGY (CONTINUED)

the action and is responsible for the execution and monitoring of the action. Example: Root cause analysis is initiated by the teams responsible for unresolved out-of-time incidents.

- **Metric:** “Quantitative measure of the degree to which a system, component, or process possesses a given attribute.”^b
- A KPI is quantified by one or more metrics. These metrics are extracted from the software system. Example: Unresolved out-of-time incidents, Total unresolved incidents
- **Measure:** “A variable to which a value is assigned as the result of a measurement.”^c It is the atomic value that is the output of a measurement process; hence, we treat it as a base measure.^{S2} Based on the context, measures can be aggregated by using a compute function to define a metric. Example: incident status, reporting time, resolution time, total incidents, etc.

The breakdown into five hierarchical concepts makes it easy to visualize and decompose a high-level goal into

^bISO/IEC/IEEE 24765:2017

^cISO/IEC/IEEE 15939:2017

concrete measurements. The rationale for these measurements is associated with the identification of a CSF and a KPI. These building blocks can be used to define a KPI in a structured manner. The process can consist of a top-down or a bottom-up approach (Figure S2).

References

- S1. S. D. Conte, H. E. Dunsmore, and Y. E. Shen, *Software Engineering Metrics and Models*. Redwood City, CA, USA: Benjamin-Cummings Publishing Co., Inc., 1986.
- S2. N. Fenton and J. Bieman, *Software Metrics*, 3rd ed., (Chapman & Hall/CRC Innovations in Software Engineering and Software Development Series). London, England: CRC Press, Sep. 2020.
- S3. V. R. Basili, G. Caldiera, and H. D. Rombach, “The goal question metric approach,” in *Encyclopedia of Software Engineering* Hoboken, NJ, USA: Wiley, 1994, pp. 528–532.
- S4. Systems and Software Engineering—Measurement Process, ISO/IEC 15939:2017, International Organization for Standardization, 2017. [Online]. Available: <https://www.iso.org/standard/71197.html>
- S5. C. V. Bullen and J. F. Rockart, *A Primer on Critical Success Factors*, Massachusetts Institute of Technology, Sloan School of Management, Massachusetts, USA, Tech. Rep. 1220-81, 1981. [Online]. Available: <https://dspace.mit.edu/handle/1721.1/1988>

be measured by the current function of the car and the future path that is visible to the driver.

- **Frequently measured:** A KPI needs to be frequently measured and continuously monitored to check its trajectory of meeting the target values.

Figure 1(c) shows the checklist items and the KPI model elements that it evaluates. For instance, the checklist helps check that both goal and CSF are detached from HR

incentives (i.e., focus on service availability rather than rewarding employees based on the number of incidents resolved).

Unraveling KPI Development in Practice

Models guide processes but do not replace human experience; their utility depends on practical use. To assess our framework’s applicability, we elaborate on the four case studies for KPI creation in practice.

Case 1: Availability KPI

The finance world is regulated through standards set by the national central bank in conjunction with (in our case) the European Central Bank. These regulations encompass various aspects of a bank’s operations, defining what needs to be measured and the desired outcomes. Among these standards lies a crucial focus on the availability of Internet banking and instant payments. This KPI takes on a unique role as an escalation limit

set by external regulators. In other words, this KPI sets a threshold above which concrete action must be taken.

Due to its significance, the availability KPI is extensively documented, monitored, and communicated across various management levels. Consequently, our interview participants easily engaged with our KPI template and defined its building blocks. However, while determining the goal and CSF, a discussion emerged concerning the origin of the goal. While the internal goal centers around meeting external regulations, the external goal, as set by the European Central Bank, emphasizes the need for stable and reliable payments for the broader society.

We can confirm that our KPI template and checklist facilitated stimulating discussions, leading to reevaluating decisions. One study participant noted, “[the template] is nice to see the big picture and for teams to see where they are in the big picture.” As technology and regulations continuously evolve, this understanding becomes increasingly crucial for software-driven companies to achieve a balance between meeting external demands and fostering internal strategies.

For “availability KPI” (for Case 1 in Table 1), the metric is translated directly as a KPI. The only difference between the two is the need for a target and an action for a KPI, as the availability metric itself can be reused for another KPI as well. The targets are divided into two threshold values based on traffic time (prime and nonprime). Although a CSF was not documented internally in the organization, the participants identified one in the context of this KPI’s goal.

Case 2: Incidents KPI

Smooth operations are of high importance for a financial institution. To ensure the availability of services, incident resolution is monitored across all services. An incident translates to an event responsible for service degradation or disruption, causing limited to no availability of services to the users.

Conformance to availability and restoration of services drive incident management goals, creating an interdependency with the availability KPI. Our study participant called it a “subprocess of availability.” Such dependencies need to be analyzed for the validity of the KPIs and the impact of their actions on other KPI values. Case 2 in Table 1 shows the “out of time incidents” KPI mapped on our template. Our study participant found it “fitting easily” to their KPI process. It measures the percentage of incidents whose resolution is due per the agreed timeline.

This KPI has a target based on the number of incidents and resolution time. Our study participant observed that “the number of incidents is not a very smart way to measure.” This number is dependent on factors like deployment frequency, number of application users, etc. which may change the perspective in terms of performance indication. Further, the resolution time may be different for different incidents and its correlation is not established yet with the incident type, priority, impact, and time-to-resolve. Such metrics may be important to measure monitoring progress in terms of incident resolution, but their representativeness of organizational performance is doubtful.

This KPI assesses incident impact on end users, especially total service disruption, determining incident

resolution priority. It is the sole KPI in our case studies considering end-user impact for incident prioritization.

Through annual reviews, tweaks were made to the KPI over time. The institute discovered that an improperly defined KPI can potentially “blur the situation” or “diversify the attention.” According to our study participant, “simplification is key.” The focus should be on the value provided by the KPI for organizational performance instead of what possible visualizations can be derived from raw data.

Case 3: Capacity KPI

In the participating financial institution, the third KPI revolves around cloud server capacity and its scaling for improved efficiency. By monitoring this KPI, the organization identifies opportunities for cost savings through recommendations. For instance, when the system detects unused resources in a four-CPU core cloud instance, it suggests resizing the server to a two-CPU core instance. The cloud provider’s cost calculator provides an estimate of cost savings per recommendation.

While this KPI may seem straightforward, we discovered that it actually aligns more with the goal of cost savings, rather than with managing cloud resources, as its name implies. Due to this observation, we recognize that our checklist could be extended by having a control check ensuring the KPI accurately represents its intended goal.

During the discussions with our study participants, all checklist items were easily verified. However, the topic of decoupling HR incentives sparked a brief discussion. The study participant shared that internal ideas included rewarding teams for outstanding performance, potentially

with nonfinancial incentives, like free pizza. While these rewards might be acceptable, the checklist inspired a discussion about keeping KPIs independent of financial incentives.

Furthermore, we uncovered ideas of reusing this KPI to achieve the goal of reducing energy consumption. Although resizing cloud components

can enhance energy efficiency, this statement cannot be universally generalized. Reusing a KPI for multiple goals may violate the checklist item “Target Driven.” If the target shifts from “right-sizing cloud applications” to “becoming more sustainable,” the KPI could lead to wrong conclusions and especially wrong actions.

The “capacity KPI” (Case 3 in Table 1) is not represented as a capacity value. Instead, it is represented as a collection of measures. Metrics are found missing. The KPI target is achieved if the recommendations for improving capacity with a cost of more than €250 are resolved within two months. If they are open

Table 1. KPIs from the case studies.

Case 1: Availability KPI					
Goal	CSF	KPI		Metric	Measure
Adhere to external regulations	Pass internal audit	Availability (%)		Availability = (No. of successful transactions / No. of total transactions) × 100	No. of successful transactions No. of total transactions
		Target	Action		
		>99.88% prime time >98.5 nonprime time	Resolve failures		
Case 2: Incidents KPI					
Goal	CSF	KPI		Metric	Measure
Ensure end-users with functionality as soon as possible without disruptions	Solve IT incidents within agreed timelines	Out of time incidents (%)		Importance Time to resolve	Incident, incident time, reporting time
		Target	Action		
		<10% Out of time incident resolutions	Root-cause analysis		
Case 3: Capacity KPI					
Goal	CSF	KPI		Metric	Measure
Cost reduction for cloud resources	Reduce the number of recommendations for cost optimization				Total recommendations, recommendation cost, recommendation time
		Target	Action		
		Recommendation cost; >€250 and open recommendation <2 months	Root cause analysis and follow up action plan		
Case 4: CO ₂ emissions KPI					
Goal	CSF	KPI		Metric	Measure
Become net-zero by 2023	Cost-effective CO ₂ reduction	CO ₂ emission			Power (kW)
		Target	Action		
			[Proposed] Optimize power settings of monitors		

for action after two months, alerts should be triggered.

Case 4: CO₂ Emissions KPI

We conducted our final case study in an aviation organization that is creating a new KPI with the premise that reducing the brightness of display screens cuts power usage. This would eventually contribute to the reduction in CO₂ emissions at the organizational level. However, there is a caveat of doing it at an optimal cost.

Case 4 in Table 1 shows the “CO₂ emission KPI” created using a bottom-up approach. A power consumption metric is identified that is used to compute a CO₂ emissions metric, which is translated directly into a KPI. The KPI is missing a target value and a tentative action is identified by the interview participants. The participants had a hard time linking the KPI to a CSF. First, because the organization does not have any CSFs defined; second, they found the distinction between the CSF and the goal a bit fuzzy. A top-down approach can help tackle such issues by first deriving appropriate CSFs for the goals that can be used to create appropriate KPIs.

The study participants acknowledged that besides power consumption, hardware procurement also contributed to CO₂ emissions, which is not factored in the KPI definition at the time. Other projects within the organization might use different measures that impact CO₂ emissions. As a result, the use of the KPI template reveals the need for additional measures to ensure a comprehensive and accurate definition of the KPI.

The team encountered challenges in establishing cost-effective targets. Although cost played a significant role in the KPI's success, it was not explicitly addressed in the KPI goals.

Notably, this pattern of cost-effective KPI targets was also observed in the capacity KPI of the financial organization. Ideally, the CO₂ emissions target should solely focus on reducing emissions, while cost reduction should have its own set of KPIs and targets. This distinction would ensure clarity in goal setting and facilitate more effective monitoring of progress in each area.

Participants noticed a gap between project-level (operational) and organization-level (strategic) goals: projects aimed for cost-effective optimization, while the organization aimed for a net-zero carbon footprint. This difference in goals contributed to the ambiguity surrounding the KPI target. Moreover, the KPI was designed to be action-driven, with plans to adjust power settings at arbitrary times to identify optimal targets. Through this exercise, the participants saw the importance of evaluating actions' impacts on KPI values and how changes, like new hardware procurement, could affect KPI measurements and targets. Aggregating the CO₂ emissions KPI into an organization-level sustainability KPI was also challenging.

Insights From the Case Studies: A Retrospective

The four case studies facilitated the evaluation of the utility of our KPI template and checklist in the context of software. They helped the organizations in both visualizing the KPI creation structure and identifying the shortcomings in the existing KPI creation and monitoring approaches. The following presents our related insights for theory and practice.

Insights for Practice

- *Monitor for inter-KPI dependency:* Software systems are

subject to coupling between different components. KPIs for software systems also reflect these interdependencies. While taking action, KPIs should be monitored for dependencies and the impact of chosen actions. The KPIs can also be used to visualize the dependencies between software quality attributes represented by such metrics.

- *Reuse measures:* With our KPI framework, we encourage reusing measures to enable a uniform and consistent measurement process. Large organizations with different departments may use the same metric in different contexts for their respective KPIs. Hence, one metric can be used in multiple KPI definitions based on the different contexts of projects in the organization. These metrics can also be reused to monitor the achievement of milestones via objectives and key results. The measurements can also be reused for other quality and progress checks within the organization.
- *Develop regulations and policies to drive KPI targets:* Goals for organizations may be divided into internal and external goals. These external goals are driven by external stakeholders, like the government or central regulatory authorities. As observed, these regulations have a significant impact on the creation of and conformance to a specific KPI, as they often come with monetary or operational penalties. The regulations may also define the exact targets and metrics. Hence, regulatory intervention can play a pivotal role in introducing new KPIs in a systematic way. As an example, emerging

technology areas, like generative AI or sustainability reporting, call for regulations. Sound KPIs can help organizations conform to such new requirements.

- *Define realistic actions at KPI design time:* To reach a KPI target, concrete and realistic action points should be defined at the design stage, addressing how to improve the KPI or respond if it falls behind. Metrics help identify suitable action areas. For long-term goals, monitoring the velocity of target achievement ensures making the goals, targets, and actions more realistic.
- *Incorporate feedback loops to manage outdated KPIs:* From our cases, we observed that annual audits are conducted to validate the relevance of a KPI. With the rapid evolution of software, periodic audits ensure the relevance of KPIs to organizational goals. This further ensures the collection and monitoring of only essential data by removing outdated KPIs. It is important to include a feedback loop involving all stakeholders of a KPI to ensure the KPIs are relevant to the organizational goals over time.
- *Avoid bottom-up KPI creation:* We recommend a top-down or hybrid approach (Figure S2) to KPI creation, ensuring organizations measure only what's necessary. A bottom-up approach risks force-fitting KPIs to goals, skewing performance representation, and causes organizations to struggle with target setting.

Insights for Theory

- *Terminological separation of concerns makes the KPI framework tangible:* Varying terms in measurement theory create ambiguity

about measures, metrics, and KPIs. Organizations often confuse these terms. Clear differentiation allows systematic aggregation of measurements into KPIs, enables reuse by keeping measures and metrics independent of targets, and prevents the definition of an excessive number of KPIs.

- *The idea of a single value target is not absolute:* The current KPI model [Figure 1(a)] was created with the assumption that a KPI has a single target with a single threshold value. From our cases, we observe that some KPIs may require multiple KPI thresholds based on contextual information, e.g., availability KPI. Other KPIs may require a composite target based on the fulfillment of certain conditions, e.g., capacity KPI. Hence, a KPI can have single, multiple, or composite targets.

Looking Ahead

Our framework helps practitioners brainstorm and refine KPIs, combining strategic and measurement elements to facilitate discussions between business and technical stakeholders. Insights from case studies translate strategic goals into actionable plans. Designing KPIs tailored to monitor the right measures allows continuous assessment and adjustment, which is crucial for maintaining compliance. This proactive approach helps decision makers stay ahead in the agile software development and regulatory landscape.

The KPI framework can be employed to represent and monitor the quality of software across any phase of the software development lifecycle with scoped goals. For instance, Sürücü et al.⁷ provide a list of KPIs to measure software development processes. They identify the challenge of

filtering the right data to create customized KPIs per organization, which can be supported through our goal-focused KPI framework by measuring metrics motivated by critical success factors. They also acknowledge stakeholder involvement as a crucial factor in the KPI definitions, which is also reflected by our case studies.

Challenges lie in the definition of KPI functions to aggregate metrics into a single KPI value. Although theory can provide models and frameworks to develop KPIs, expert experience and knowledge must supplement the process for sound KPI development, e.g., by keeping the number of KPIs small, defining abstraction levels, providing theoretical support for KPI function definitions, settings targets, etc.

Looking ahead, measurement frameworks (e.g., Sustainability Assessment Framework Toolkit⁸, Green Software Measurement Model⁹) or awareness frameworks (e.g., Sustainability Awareness Framework¹⁰) offer the potential to integrate our template and checklist to create sound KPIs. Follow-up case studies may explore these frameworks in conjunction with our template and introduce new checklist items as needed. In the long term, we will benefit from widely accepted and well-defined KPIs that facilitate the measurement of software systems and improve their comparability.

We evaluated the utility of the proposed KPI framework via four case studies and gained insights into the dos and don'ts of creating KPIs. The study participants found the framework helpful, intuitive, and actionable. In the future, we aim to use the instruments of this KPI framework to facilitate software



IFFAT FATIMA is a Ph.D. candidate with the Software and Sustainability Group (S2) at Vrije Universiteit Amsterdam, 1081 HV Amsterdam, The Netherlands. Her research interests lie at the intersection of software architecture and sustainability. Fatima received her M.Sc. in software engineering from the National University of Sciences and Technology in Pakistan. Contact her at i.fatima@vu.nl.



MARKUS FUNKE is a Ph.D. candidate with the Software and Sustainability Group (S2) at Vrije Universiteit (VU) Amsterdam, 1081 HV Amsterdam, The Netherlands; and a researcher at the VU Digital Sustainability Center, the Sustainable IT Lab, and the EU research project μ DevOps. His research interests include software architecture, architecture knowledge, and software sustainability, with particular interest in the connection to professional practice and the industry. Funke received his joint M.Sc. in computer science from the Vrije Universiteit Amsterdam and the University of Amsterdam, The Netherlands. He is a Graduate Student Member of IEEE. Contact him at m.t.funke@vu.nl.



PATRICIA LAGO is a professor in software engineering at Vrije Universiteit Amsterdam, 1081 HV Amsterdam, The Netherlands. Her research interests include software architecture design and decision making, software quality assessment, and software sustainability. Lago received her Ph.D. in control and computer engineering from Politecnico di Torino. She is a Member of IEEE. Contact her at p.lago@vu.nl.

practitioners in identifying CSFs from software quality goals. Supplemented with the available measurement frameworks, this will aid in identifying the right metrics for action-driven KPIs. 

Acknowledgment

This publication is part of the project SustainableCloud (OCENW. M20.243) of the research program Open Competition which is (partly) financed by the Dutch Research Council (NWO). We thank all

interview participants^a from our industrial partners who participated in this study and donated their time and knowledge.

References

1. P. F. Drucker, *Management: Tasks, Responsibilities, Practices*. New York, NY, USA: Harper & Row, 1973,
2. R. S. Kaplan and D. P. Norton, *The Balanced Scorecard: Translating*

^aInterviewees and data are kept anonymous for privacy reasons.

- Strategy Into Action*. Cambridge, MA, USA: Harvard Univ. Press, 1996.
3. D. Parmenter, *Key Performance Indicators: Developing, Implementing, and Using Winning KPIs*. Hoboken, NJ, USA: Wiley, 2015.
4. G. E. P. Box, "Science and statistics," *J. Amer. Statist. Assoc.*, vol. 71, no. 356, pp. 791–799, 1976, doi: [10.2307/2286841](https://doi.org/10.2307/2286841).
5. G. E. P. Box, N. R. Draper, and R. W. Hooper, *Empirical Model-Building and Response Surfaces*. Hoboken, NJ, USA: Wiley, 1987.
6. G. T. Doran, "There's a S.M.A.R.T. way to write management's goals and objectives," *Manage. Rev.*, vol. 70, no. 11, pp. 35–36, 1981.
7. C. Sürücü, B. Song, J. Krüger, G. Saake, and T. Leich, "Establishing key performance indicators for measuring software-development processes at a large organization," in *Proc. 28th ACM Joint Meeting Eur. Softw. Eng. Conf. Symp. Found. Softw. Eng.*, (ESEC/FSE), New York, NY, USA: Association for Computing Machinery, 2020, pp. 1331–1341, doi: [10.1145/3368089.3417057](https://doi.org/10.1145/3368089.3417057).
8. P. Lago, N. C. Fernandez, I. Fatima, M. Funke, and I. Malavolta, "The sustainability assessment framework toolkit: A decade of modeling experience," 2024, *arXiv:2405.01391*.
9. A. Guldner *et al.*, "Development and evaluation of a reference measurement model for assessing the resource and energy efficiency of software products and components—Green software measurement model (GSMM)," *Future Gener. Comput. Syst.*, vol. 155, pp. 402–418, Jun. 2024, doi: [10.1016/j.future.2024.01.033](https://doi.org/10.1016/j.future.2024.01.033).
10. B. Penzenstadler *et al.*, "Vision paper: The sustainability awareness framework (SusAF) as a de-facto standard?," in *Proc. CEUR Workshop*, vol. 3378, 2023, p. 4.