Framework for creating relevant, accessible, and adoptable KPI models in an Industrial Setting

Arttu Leppäkoski, Outi Sievi-Korte and Timo D. Hämäläinen

Tampere University, Tampere, Finland
arttu.leppakoski@tuni.fi
outi.sievi-korte@tuni.fi
timo.hamalainen@tuni.fi

Abstract. The development of software for modern products with lots of interfaces, layers and stakeholders has become very complex, increasing the risk of inefficiency. Key Performance Indicators (KPIs) can be used to identify bottlenecks and problems, but the challenge is how to create KPI models, processes and dashboards that help improving the development processes and can be adopted by all the stakeholders. We introduce the RelAA Framework - a bottom-up approach for monitoring product-focused software development. The RelAA (Relevant, Accessible and Adoptable) Framework is created in an industrial setup that currently includes around 350 persons in different phases of the software life cycle. The RelAA Framework is formed by analyzing existing KPIs and tools, gathering feedback from development teams, management, business representatives, and other stakeholders, and creating intuitive ways to share information related to KPIs. The RelAA Framework itself does not define exact KPIs for the organization to adopt, but it provides a process and model how to create, document and utilize KPIs. The RelAA Framework ensures relevance, accessibility, and adoption of KPIs across stakeholders and organization.

Keywords: KPI, KPI Model, KPI Framework, Metrics, Software Life Cycle, Dashboard, Visualization, Agile.

1 Introduction

Large organizations have increasingly adopted performance measurement programs to aid decision-making and control quality. Key Performance Indicators (KPIs) are often at the center of these programs, as they are the concrete measures used to quantitatively assess performance of critical factors [1]. However, companies may struggle with creating pro-active measures and selecting the right set of KPIs for long-term analyses, among other issues [2]. Increasing software complexity and development team sizes set further limitations and new prerequisites for the KPIs. In most cases, both the KPIs and goals have been defined and given “as is” by the management. This can lead to some of the KPIs not having clear linkage to the daily work of the development teams and can even be considered confusing or disturbing. In the worst case, the KPIs that are designed to be used by only certain types of teams are used by other teams as well. This

increases the risk of teams not committing or taking actions to improve their work, and thus, KPIs not being optimally exploited in optimizing the overall operations. We thus set the following research questions:

RQ1: Relevance - *How to ensure that the used set of KPIs is relevant, up-to-date and focuses on the right context to the stakeholders and teams?*
RQ2: Accessibility and adoption - *How to ensure KPIs are seamlessly accessible and a natural part of the daily work?*

Via an action research process to answer these questions, we propose the RelAA (Relevant, Accessible and Adoptable) Framework which consists of RelAA Process, RelAA Model and RelAA Dashboard. We suggest a process where the KPI model is developed bottom-up and provide facilitation for the creation and management of the model with web-based tooling. Instead of defining and offering a fixed KPI model, we introduce a process to engage stakeholders to contribute to the model, which is also expected to evolve according to the needs, relevance, and usefulness to each stakeholder group. In addition, the framework provides methods to ensure that company business model and targets are aligned with KPIs. Piloting the framework in our case company has shown promising results on rising commitment to KPIs.

The rest of the paper is organized as following. Section 2 discusses the background for this research. We first go through the related work on KPI models and processes, and then introduce the industrial background. Section 3 describes the utilized action research process, resulting in an introduction of the RelAA Framework within the case study company. Section 4 introduces the RelAA Framework. Section 5 evaluates the completeness of the RelAA Framework. Section 6 includes lessons learnt and lists possible threats and weaknesses. Section 7 concludes the paper and introduces ideas for the future work.

2 Related Work and Background

2.1 Related Work

Performance measurement is a widely studied phenomenon in a variety of business domains (see., e.g., [3, 4, 5, 6]). Utilizing KPIs is an inherent part of performance measurement. In this study, our scope is on KPIs targeted particularly for software development, which is distinctly different from, e.g., manufacturing or construction. There is scarce research on the processes for creating KPI models, utilizing KPIs and on the actual KPI models targeted specifically for SW engineering processes. While SW *metrics* and measures are widely used and discussed, an *indicator* is more complex, being comprised of several measures [7]. For example, Briand et al. [8] propose the GQM/MEDEA method - a general process that can be used as a guideline to design sound measures particularly in the field of SW engineering. However, more steps would be needed to further refine such measures into KPIs.

There are some studies suggesting the processes to define KPI models in the context of software development. Tsunoda et al. [9] present a general model for discussing SW projects and their success with a variety of stakeholders, and KPIs or Key Goal

Indicators (KGIs) are an essential part of that model. In their paper, a mock example of a KPI/KGI model is presented, based mainly on counting defects and program size. Their main contribution is a set of requirements that need to be met when designing a model to enable communication between stakeholders.

Staron et al. [10], in turn, present a KPI Quality model, defining 59 quality attributes that a well-defined KPI needs to meet. The attributes are based on three key drivers: Transparency, Actionability, and Traceability. Additionally, Staron and Meding [11] present the MeSRAM method for assessing robustness of measurement programs particularly in SW development. While MeSRAM considers measurement programs at large, KPIs are an essential part of it.

There are some studies giving examples of KPI models in software engineering. Antolic [12] presents a KPI model for SW development. While the KPIs are based on data and experiences from Ericsson Software Development, the process of how the KPI model was established and how it has evolved has not been enclosed.

Kazi et al. [13], in turn, use a balanced scorecard approach for monitoring performance in SW projects, where KPIs make the scorecard. While the KPIs are based on data stores used in the business process model, and the KPI model is carefully mapped to serve existing process models, there is no report on how the KPI model would evolve, and what kind of practical experiences there are in using the model.

Finally, a review shows [14] that there is little research on visualizations related to SW processes, including visualizing KPIs alongside other process information, as most SW visualization studies focus on visualizing programs.

According to our study, we find the gaps in existing research as listed in Table 1. We will contribute to filling in these gaps. Proposed solutions are linked to the gaps in the later parts of this paper.

Table 1. Gaps in the existing studies

ID	Description	Research question
GAP1	How are teams included in the process of defining KPIs?	RQ1
GAP2	How to create a set of KPIs that are applicable for monitoring a larger entity than just one SW project?	RQ1
GAP3	How to create a set of KPIs that are targeted for different stakeholders?	RQ1, RQ2
GAP4	How to support evolution in KPIs?	RQ1

2.2 Case company background

The research in this paper sparked from acute needs in a large multi-national company, providing industrial products in all continents. The products include mechanics, electronics and both embedded and cloud scale software. Software is involved in controlling mechanical devices, connecting millions of devices to cloud, executing cloud analytics and delivery of user interfaces for different stakeholders.

The number of persons working in the SW department from 2013 to 2021 has increased from 75 to more than 300 and the development sites (USA, India, and Finland)

are now more geographically distributed than before. The growth requires both new organizational structures and development processes.

The SW department includes three different programs which include tens of Scrum teams. Scrum teams have two-week sprints starting with sprint planning, having daily standups, and ending to sprint demo session. Programs and Scrum teams plan their backlogs for the next three months in quarterly planning sessions together with the business and other stakeholders such as maintenance. Quarterly planning is used to align the backlogs and schedules according to the given business priorities and make sure that possible dependencies to the HW development is aligned.

In the case company, the main purpose of KPIs is to ensure that all activities and initiatives are providing additional customer value and that operations are carried out effectively. KPIs can be either strategic or operational and both short-term and long-term targets can be set. KPIs are used in different levels of the organization and for different target audience and roles.

3 Research process

Our research process follows the action research [15] method which includes four phases: plan, act, observe, and reflect. The design and development of the RelAA Framework has been carried out in phases as depicted in Fig. 1.

Phase	Action research	2018	2019	2020
SW development life cycle assessment	Plan			
Planning phase	Plan			
Development program	Not applicable			
Identification of the existing KPIs and tools	Plan			
Definition of the RelAA Framework and new KPIs	Act			
Feedback gathering (workshops & discussions)	Observe, reflect			
Development of the Minimum Viable Product (MVP)	Act			
RelAA Framework launch	Act			
Feedback gathering (RelAA Dashboard & discussions)	Observe, reflect			
Continous development	Observe, reflect			

Fig. 1. The action research phases and timeline in the RelAA Framework development

3.1 Planning phase and initial observations

The planning phase included schedule setting, stakeholder identification, analysis of the existing KPIs and processes, and goal setting. All the major stakeholders have been involved in the process: SW teams, HW teams, management, business, quality managers, SW testing department, steering group, and Human Resources (HR). The planning phase included an analysis of the characteristics of the existing KPIs, tools and processes, as well as several workshops and face-to-face discussions to gather feedback. The following questions were applied in the workshops and discussions:

- Do you understand why we have KPIs?
- Are you able to find required documentation for each KPI?

- Have you been able to use the existing KPI tool?
- Which KPIs and metrics have been useful?
- What kind of things would you like to measure in your team?
- What would help your team to get more committed on KPIs?

Workshops, discussions, and interviews included persons from different parts of the organization and different roles such as program managers, developers, test specialists, product owners, agile coaches, quality managers, and test managers. Table 2 includes a summary of the feedback gathered from the workshops and interviews grouped into strengths, weaknesses, opportunities, and threats. For example, the interviews revealed that only 10 out of the 20 persons interviewed considered the existing KPIs useful and that KPIs had been guiding teams into better performance. Thus, there was a clear need to improve the existing model and its adoption. The first draft of the framework was offered for stakeholders' evaluation and further evolution based on their contributions.

Table 2. SWOT analysis

Strengths • Lots of existing KPIs for teams and programs • Automatic system to gather data from existing systems (e.g. backlog management) • Story point burnup charts are widely used by development teams and visible in info screens • Lots of different kind of visualizations and chart types available in the existing tool • KPIs are used for measuring the performance of development teams instead of individuals
Weaknesses • Lack of documentation, only small set of KPIs have documentation available • Ambiguity and obscurity of KPIs • Disorganized and unintuitive user interface and no means to enter data points manually • Missing linkage between the KPIs and objectives defined by the management • Missing linkage between the KPIs and company processes • No KPIs for measuring customer value and other levels than team and program • Lack of SW testing related KPIs • Goals are not understandable and justifiable by the development teams • Only part of the KPIs have goals in place
Opportunities • New KPIs for other levels than team and program • Definition of KPIs and goals for each role (Product Owner, Scrum master etc.) separately • Soft KPIs to measure motivation and feelings of the organization • Appropriate balance between agility and stability to increase commitment • Automatization of measurement of new KPIs
Threats • Insufficient quality of data • Incorrect and misleading goals and results • Team adjusts daily routines to meet the goals without understanding the reasons • Lack of common goals due to the diversity of teams • Usage of KPIs as incentives arouses controversy in the organization • Negative attitude towards KPIs • Excessive amount of KPIs • Excessive and complex documentation

3.2 Acting phase

Acting phases include defining the process for continuous KPI development, setting guidelines for the documentation of the KPIs and implementation of the facilitation tool. Development of these entities is tied to a larger company-wide program, dedicated for improving the SW life cycle processes in an industrial setting. The outcome of the acting phase, RelAA Framework, is described in Section 4. One of the authors is working in the program within the company. The program started in 2018 and completed at the end of 2020. A dedicated info session for the teams involved in the SW development life cycle was arranged in June 2019 to launch the RelAA Framework. The implementations of an automatic data gathering system, a simple KPI tool and a set of KPIs for automated testing and continuous SW integration was started by Oinonen [16].

3.3 Observe and reflect

Development of the RelAA Framework was followed in a weekly program status meeting in which the program core team provided their observations and guidance. Development of the RelAA Framework was coordinated, and observations were discussed in regular sessions with the RelAA development team. Additions to the RelAA Framework were reviewed and discussed in dedicated sessions with the program core team and selected stakeholders. For example, new RelAA Dashboard views, changes to the RelAA Process, and additions to the RelAA Model were demonstrated to the program core team and stakeholders to collect feedback. In addition, feedback was gathered from the RelAA Dashboard users by providing a built-in feedback form and having informal discussions with many people. With methods above we gathered observations continuously throughout the iterative development of the RelAA Framework. Observations were reflected to adjust and improve the RelAA Framework iteratively.

4 RelAA Framework

RelAA Framework consists of RelAA Process, RelAA Model and RelAA Dashboard which are tightly integrated with each other to provide seamless user experience. Fig. 2 describes the characteristics of these components. *RelAA Dashboard* is a facilitation tool that supports the creation, usage and analysis of the RelAA Model and RelAA Process. *RelAA Model* defines guidelines and templates for documenting the KPIs for better visibility and transparency. *RelAA Process* defines processes and methods to enable the continuous development of the KPIs.

4.1 RelAA Process

KPIs require continuous development due to the changing development processes, organizational structures, and evolving architectures. Thus, it is important to have proper methods in place to ensure that the KPI model remains in good shape. In this section we introduce the RelAA Process, providing steps for 1) managing the KPI life cycle,

2) setting clear roles and responsibilities, 3) enabling the incremental development, 4) validating the defined KPIs and 5) providing training for the end users.

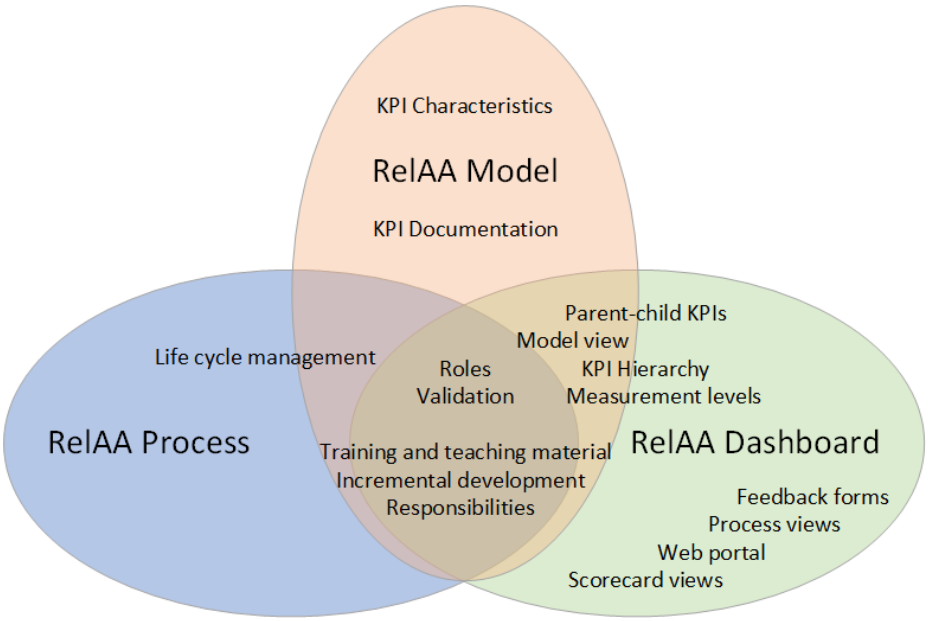

Fig. 2. RelAA Framework

Life cycle management. The purpose of the RelAA Process is to allow teams and stakeholders to be involved in the KPI specification from the beginning and thus increase the adoptability and transparency of KPIs, as well as and commitment to them. Additionally, there was a need to identify KPIs that are not seen valuable by the users. This approach requires that each KPI has a predefined status as listed in Table 3. Each status phase includes various tasks that need to be carried out before the next phase can be reached. The KPI life cycle and roles and responsibilities are illustrated in Fig. 3.

Table 3. Status values

Status	Description
Define	Definition of KPI and its characteristics (like measurement criteria) is in progress. Changes may occur during this phase.
Baseline	KPI and its characteristics have been defined. Data is being collected to set the baseline and targets. Measurement results and measurement criteria are being validated by relevant stakeholders, teams, and persons.
Measure	KPI and its characteristics have been defined and baseline and targets have been set. KPI is being followed by the responsible person or team.

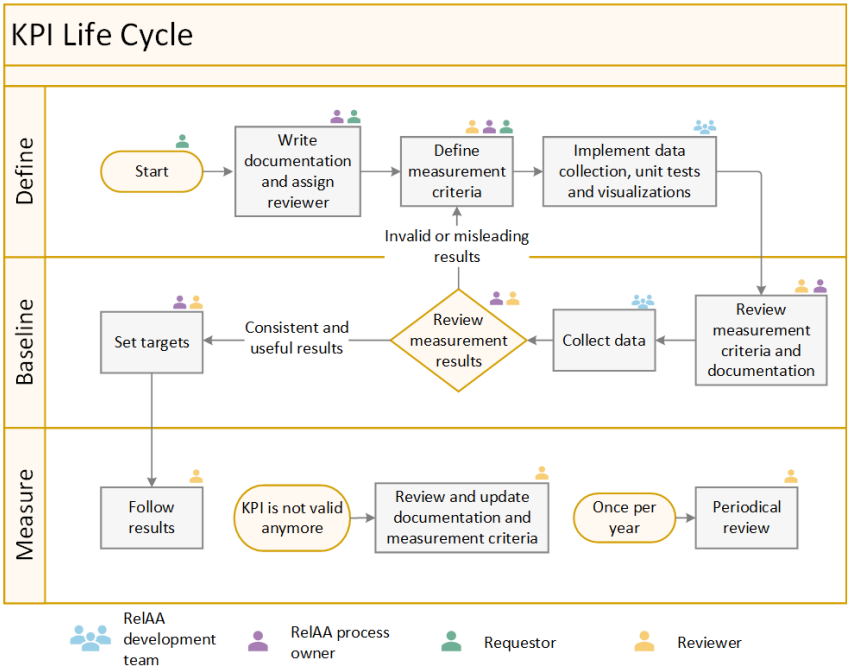

Fig. 3. KPI life cycle

Roles and responsibilities. One of the key targets was to set clear roles, responsibilities, and accountabilities to ensure that KPIs are being developed and followed continuously. We have defined following roles and responsibilities.

- *RelAA Development team* is responsible for implementing data collection, dashboards, and visualizations. Team consists of developers that implements and maintains the system that is gathering data from different sources and components required in the RelAA Dashboard. Development team owns the RelAA Dashboard.
- *RelAA Process owner* is responsible for creating roadmaps and schedules for implementation of new KPIs, collecting ideas and feedback, and participating to KPI documentation reviews. Process owner interacts with the RelAA development team actively. Process owner owns the RelAA Process and RelAA Model.
- *Requestor* is a person or team that proposes a new KPI to be taken into use or existing KPI to be modified.
- *Reviewer* is a person that is responsible for following the measurement results, setting targets, and checking that the measurement criteria and documentation is up to date. Reviewer is assigned for each KPI separately.
- *End user* is a person or team that uses the RelAA Framework to follow the KPIs.

Incremental development. Sometimes it is required to modify existing KPIs, introduce a new KPI or completely stop using some KPI. To enable the required incremental

development process, we decided to include version numbers and histories to both the KPI model and RelAA Dashboard. This should allow comparison of different versions and provide visibility on how KPI model and RelAA Dashboard change and evolve.

Several reasons (triggers) can cause a new version of the KPI model to be released. Table 4 includes triggers that we have identified to result in releasing a new version. For each trigger, we have defined a set of actions to be executed. Predefined actions are targeted to help ensuring that KPI model remains up to date in all circumstances. In addition, we have defined an update interval for the KPI model. The purpose of the update interval is to provide information and visibility about the pace and scope of changes in relation to the nature of changes. The KPI model can be updated weekly, monthly, or yearly (Table 5).

Table 4. Triggers for new version of KPI model

Trigger	Description	Actions
Process change	Many KPIs are usually related to process milestones or phases.	Review process views Review documentation
Organizational change	Changes in the organization may cause changes in responsibilities and teams.	Assign reviewers Review measurement levels
Architectural change	KPIs may become obsolete or invalid due to architectural changes.	Review measurement criteria Assign reviewers
Invalid goals	Goals are not realistic or have been achieved.	Review Target and Stretch goals
Invalid or insufficient results	Poor quality of the measurement results. KPI does not provide any additional value in finding problems or bottlenecks.	Review measurement criteria Review data quality
Development tool set change	New tool or new version of existing tool is taken into use.	Review data sources Review measurement criteria
KPI missing	New KPI has been identified	Add KPI into the KPI model

Table 5. KPI model update interval

Interval	Description of changes	Examples
Weekly	Minor changes that do not have long-term impact or do not require any actions from teams or stakeholders	Documentation improvements Bug fixes
Monthly	Medium changes that may require some minor actions from teams or stakeholders	Changes to measurement criteria Changes in reviewers Adding of new KPIs
Yearly	Major changes that must be communicated to whole organization	Setting of goals used for incentives Process view updates

Validation measures. To ensure the validity of the KPIs through the whole KPI life cycle, we have defined the following validation measures.

Validity of new KPI. RelAA Development team together with the reviewer and relevant stakeholders validates the measurement results and reviews the documentation of the new KPI during the Baseline phase.
Validity of existing KPIs. Each KPI is being reviewed (documentation and measurement results) periodically at least once per year to ensure that the underlying data is in good shape and reliable and documentation corresponds with the actual ways of working. Reviewer is responsible to execute the periodical review.
Validity of data and visualizations. RelAA Development team creates the necessary unit tests and other tests to ensure that the measurement results are being calculated and visualized correctly. Tests are executed automatically always when changes occur.

Training sessions and teaching material. During the development of the RelAA Framework we discovered that even if the tools and processes would be intuitive and documentation is available, separate training sessions and teaching material need to be arranged. Developing the RelAA Framework we defined training sessions for different target audiences and added a description of the RelAA Process and a dedicated help view to the RelAA Dashboard.

Training sessions are tailored for each role (such as Product Owner), but each training session has a common frame including two parts (2 hours). The first part focuses on basics and theory and includes description of the RelAA Framework. Second part is a hands-on session in which participants are getting familiar with the RelAA Framework by trying it by themselves.

4.2 RelAA Model

In the RelAA Model, KPIs are organized hierarchically (parent-child) to increase the accessibility and adoptability and to illustrate relationships. The value of the parent KPI is calculated using the values of the child KPIs using pre-defined measurement criterium. Fig. 4 includes a small exemplary set of KPIs and their hierarchical relationships.

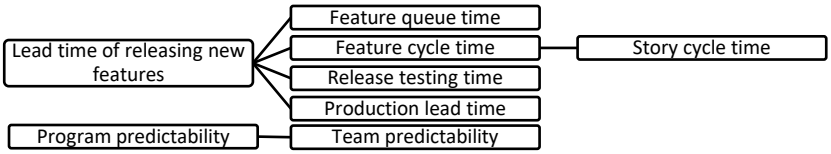

Fig. 4. Hierarchically organized KPIs

The RelAA Model includes eight documentation fields (free-form text) that are used for documenting each KPI as listed in Table 6. The purpose of these is to elaborate the background, provide practical examples and to increase the acceptance and commitment in teams.

KPIs are divided into measurement levels based on the level in which a KPI is used. Each KPI is mapped to one or more measurement levels. The measurement levels are listed in Table 7.

Table 6. Documentation fields

Documentation	Description
Why is this needed?	Description about the background and purpose of the KPI. ***Example*****:** *Cycle time indicates how fast team delivers new features. Fast and consistent delivery of stories is an indicator of high productivity.*
Measurement criteria	Equation or other description on how KPI value is being measured. ***Example*****:** *Story cycle time is the time (days) each story spent in 'In Progress' state before it is moved to 'Done' or 'Accepted' states.*
Target goal	Target goal is defined only for KPIs that can have some meaningful target. Goal is set only after baselining period is completed. Target goal is set so that it is easier to achieve than stretch goal. ***Example*****:** *Story cycle time < 10 days*
Stretch goal	Stretch goal is defined only for KPIs that can have some meaningful target. Goal is set only after baselining period is completed. ***Example*****:** *Story cycle time < 5 days*
Reviewer	Person in charge of reviewing the results. Ideally, only one person should be nominated to avoid shared responsibilities. ***Examples*****:** *Product Owner, Program Manager, Quality Manager*
Data source	List of tools, applications and systems from which data is retrieved. ***Examples*****:** *Version control system, issue and task tracking system*
Measurement frequency	How often KPI is being measured. ***Examples*****:** *daily, weekly, monthly*
Reporting frequency	How often KPI is being reported. ***Examples*****:** *daily, weekly, monthly*
Unit of measurement	Unit of the KPI value ***Examples:*** *days, story points, number of defects*

Table 7. Measurement levels

Measurement level	Description
North Star	Overall status in company and product level
R&D	Overall status in R&D level
Solution	KPIs for getting more detailed information about specific solutions
Release	Detailed status and progress of each SW release
Program	KPIs for each SW development program
Team	Detailed status of the performance of each team
Component	Detailed status of the development of each component

Each KPI is mapped to one category and one or more SW life cycle phase based on the characteristics. The categories are Speed, Efficiency, Customer centricity, and

Motivated & capable team. SW life cycle phases are Planning, Requirements, Development, Testing, Maintenance. In addition, Each KPI is also tagged either Lagging or Leading. Lagging KPI is for measuring and analyzing past performance and Leading KPI is for measuring and predicting future performance.

4.3 RelAA Dashboard

The development of the RelAA Model was tightly coupled to the RelAA Dashboard which is a web portal user interface to the KPIs. We decided to build documentation into the Dashboard as there must be a way to intuitively explore the concrete KPI model (built according to the RelAA Model).

The RelAA Dashboard includes a dedicated *Model view*, which contains a built-in tree-view to visualize the hierarchies, and a possibility to see all the documentation related to the KPI by selecting any KPI.

Process view is used to map each KPI to the development process and the RelAA Dashboard includes several process views. Available process views are North Star, Release, Program, and Team, which have been defined as measurement levels. Colors (green, yellow, red) indicate the status of the KPI in each process view. Status is defined according to the Target and Stretch goals. The Trend is shown using arrows (up, right, left). Process views are kept rather simple to easy understanding.

Scorecard view includes documentation (items listed in chapter 4.2), charts, and values related to the selected KPI. All this information is shown in a single view to increase the accessibility of the documentation and adoption of KPIs. In addition, the Scorecard view includes dedicated discussion panel that is intended for discussing about KPI results, anomalies identified in the KPIs etc.

Feedback view includes a built-in form to give feedback about the KPI model and RelAA Framework. Feedback is made visible for all allowing users to browse given feedbacks and check which feedback has been already processed and in which version it has been taken into consideration.

5 Results

5.1 Observed effect

As part of the action research process, the RelAA framework was actively piloted in the case company while it was being created. Based on the feedback gathered during this research and development of the framework, most of the team members feel positive about the RelAA Framework and KPIs after they have been involved. They understand the background, measurement criteria and linkage to the daily work. The RelAA Framework has increased the accessibility and adoptability of the KPIs by providing intuitive, approachable, and understandable means to explore the model itself and analyze the performance of the organization.

Since the framework was completed, many of the KPIs introduced using the RelAA Framework have proven to be very useful. For example, *Engagement & Satisfaction*

KPI provides detailed insights about the motivation of the development teams, *Team Predictability* KPI indicates whether it is realistic to expect the business commitments to be fulfilled and *Story Definition of Ready* KPI depicts if teams are having enough information available before starting stories. All the KPIs together help to ensure that the teams are proceeding according to the business targets.

5.2 Fulfilling requirements

Tsunoda et al. [9] have defined a set of requirements that need to be met when designing a model to enable communication between stakeholders. These requirements are in line with our research questions and objectives, and thus, provide a useful method to validate the completeness of the current RelAA Framework. The requirements and comparison to the RelAA Framework are listed in Table 8. Requirements and models introduced in other studies were not seen comparable due to the differences in the applicability of the models. Other models are more narrowly scoped for smaller SW projects whereas the RelAA Framework is applicable for a larger entity.

Table 8. Evaluation of requirements set by Tsunoda et al. [9]

Requirement by Tsunoda et al.[9]	RelAA Framework
(R1) Goals of each stakeholder separately, and a metric which directly indicates whether the goal is achieved or not.	Yes
(R2) Distinction between a metric indicating goal achievement and metrics indicating progress toward the goal.	Yes
(R3) How to collect metrics.	Yes
(R4) How to analyze metrics.	Yes
(R5) Timing of analyzing metrics with stakeholders.	Yes
(R6) Countermeasures to correct abnormal process or products identified on check phase.	Yes

5.3 Fulfilling gaps

In Table 1 we introduced gaps that were found in the existing studies and in Section 4 we have introduced solutions to fulfill these gaps. Table 9 includes summary about gaps and our solutions.

Table 9. Evaluation of gaps

ID	Description	Proposed solution
GAP1	How are teams included in the process of defining KPIs?	RelAA Process: KPI Life cycle RelAA Process: Incremental development RelAA Process: Training sessions RelAA Dashboard: Feedback database
GAP2	How to create a set of KPIs that are applicable for monitoring a larger entity than just one SW project?	RelAA Model: Measurement levels RelAA Dashboard: Process view
GAP3	How to create a set of KPIs that are targeted for different stakeholders?	RelAA Model: Measurement levels RelAA Dashboard: Process view
GAP4	How to support evolution in KPIs?	RelAA Process: KPI Life cycle RelAA Process: Incremental development RelAA Dashboard: Feedback RelAA Dashboard: Discussion section

6 Discussion

6.1 Lessons learnt

Developing the RelAA Framework gave an intense insight into how KPIs are utilized within a large, multi-national company. Getting different stakeholders committed to KPIs required continuous discussions, active inclusion, and many training sessions. When taking the RelAA Framework to use, we particularly advice to take note of reserving a great deal of time and resources to involve all the necessary stakeholders to have all the aspects covered in the defined KPI model. It is also important to notice that something that fits one team, might not be suitable for some other team. For this reason, it is not reasonable to expect every team to follow the exact same set of KPIs and targets. Finally, we will revisit the research questions.

RQ1: Relevance - How to ensure that the used set of KPIs is relevant, up-to-date and focuses on the right context to the stakeholders and teams?

First, one should **engage** different stakeholders from the beginning of the KPI process. There should be **life cycle management** of the KPIs with distinct **roles** and **responsibilities**, and the organization should further allow **incremental development** of KPIs. The KPIs should also be **validated**, and **training material** provided for different users. The proposed RelAA process captures these elements to ensure that KPIs are up-to-date and that they are both designed to be relevant and understood as such by different stakeholders. Relevance is further improved by **comprehensive documentation**, describing the context and purpose of KPIs (as proposed by the RelAA Model).

RQ2: Accessibility and adoption - How to ensure KPIs are seamlessly accessible and a natural part of the daily work?

A key part in having accessible KPIs is to provide **relevant information** for **a variety of stakeholder** with **technical solutions** that are easy to adopt. Our solution for this is the constructed RelAA Dashboard. With the Dashboard different stakeholders can

view KPIs, **suggest** new ones, **give feedback**, and **monitor** KPI values. Particularly, by providing **a number of visualizations** for different purposes and stakeholders, we have found that KPIs are more easily taken aboard in the organization. However, in order to have the Dashboard taken in as part of daily work, **a working process** for introducing it on an organizational level is required. The RelAA Process defines that stakeholders must be engaged from early on in defining the KPIs.

6.2 Threats to validity

Using the classification presented by Wohlin et al. [17] we identified the following threats to validity.

Construct validity and Internal validity. We recognize that our view on how KPIs are being interpreted and understood is dependent on successfully selecting a varied set of interviewees and accurately constructing the interview questions. To minimize this risk, we have had discussions with several different types of teams to identify the distinction between common issues and team specific issues. In addition, the RelAA Framework has been reviewed with selected stakeholders during the development phase. The complexity of the development process increases the risk of not finding the causal relationships that are required to ensure the quality of KPIs. This risk has been minimized by defining several measurement levels and mapping the KPIs to process views.

External validity. The RelAA Framework has been used only in one organization so far, and thus, we do not have evidence of the RelAA Framework's suitability in other organizations. However, we consider this as a low risk since the organization is rather large and includes a large variety of products, teams, and stakeholders. Therefore, the RelAA Framework by nature supports diverse products and processes.

Conclusion validity. Risk of low conclusion validity exists due to the lack of numerical and statistical evaluation. This risk has been minimized by evaluating the requirements set by Tsunoda et al. [9] and reviewing the results and conclusions with the stakeholders.

7 Conclusions

We set out to create a methodology for flexibly defining KPIs for monitoring product-focused software development. Our aim was to ensure relevance (RQ1) and accessibility and adoption (RQ2) of KPIs across stakeholders and organization.

Following an action research-based process, we introduced the RelAA Framework defined for monitoring SW life cycle in an industrial setup. The RelAA Framework has been taken into use in a large company including 350 persons in SW development and in total of 56 KPIs have been documented and categorized using the methods introduced in the RelAA Framework. The defined KPI model is being updated constantly using the methods described in this paper. The RelAA Process, Model and Dashboard have been designed to fulfill the gaps (GAP1-GAP4) described in Section 2.

We will 1) continue arranging surveys about the experiences, 2) raise the level of commitment to SW process metrics by using regular workshops, info letters, and

allowing teams more control over KPIs, and 3) validate of the KPIs used in the presented industrial setup using the KPI Quality Model [10] and MesRAM method [11].

References

1. Sinclair, D. and Zairi, M., "Effective process management through performance measurement: Part III-an integrated model of total quality-based performance measurement", *Business Process Re-engineering & Management Journal*, 1 (3), 1995, pp. 50-65.
2. Staron, M.: Critical role of measures in decision processes: managerial and technical measures in the context of large software development organizations, *IST* 54, 2012, 887-889.
3. Neely, A., Gregory, M. and Platts, K., "Performance measurement system design: A literature review and research agenda", *Int. J. Oper. Prod. Man*. 15 (4), 1995, pp. 80-116.
4. Gosselin, M., "An empirical study of performance measurement in manufacturing firms", *Int. J. Product. Perform. Man.*, 54 (5/6), 2005, pp. 419-437.
5. Sanchez H, Robert B. “Measuring Portfolio Strategic Performance Using Key Performance Indicators”. *Project Management Journal*. 41(5), 2010, pp. 64-73.
6. Ishaq Bhatti, M., Awan, H.M. & Razaq, Z. The key performance indicators (KPIs) and their impact on overall organizational performance. *Qual Quant* 48 2014, pp. 3127–3143.
7. Garcia, F., Bertoa, M.F., Calero, C., Vallecillo, A., Ruiz, F., Piattini, M., and Genero, M.: Towards a consistent terminology for software measurement, IST 48, 2006, 631-644.
8. Briand, L. C., Morasca, S., and Basili, V. R.: An operational process for goal-driven definition of measures, *TSE* 28 (12), 2002, 1106-1125.
9. Tsunoda, M., Matsumura, T. and Matsumoto, K.: Modeling Software Project Monitoring with Stakeholders, In: Proc: *ICCIS* 2010, pp. 723-728.
10. Staron, M., Meding, W., Niesel, K., Abran, A.: A Key Performance Indicator Quality Model and Its Industrial Evaluation, In: Proc. *IWSM-Mensura*, 2016, pp. 170-179.
11. Staron, M. and Meding, W.: MeSRAM – A method for assessing robustness of measurement programs in large software development organizations and its industrial evaluation, *JSS* 113, 2016, 76-100.
12. Antolic, Z.: An example of Using Key Performance Indicators for Software Development Process Efficiency Evaluation, Ericsson Nikola Tesla R&D Center, 2008, 6p.
13. Kazi, L., Radulovic, B., and Kazi, Z.: Performance Indicators in Software Project Monitoring: Balanced Scorecard Approach, In: Proc. SISY, 2012, 19-24.
14. Mattila, A-L, Ihantola, P, Kilamo, T, Luoto, A, Nurminen, M & Väätäjä, H Software visualization today - Systematic literature review. In: Proc. of the 20th International Academic Mindtrek Conference. 2016, ACM, pp. 262-271.
15. Dickens, L., Watkins, K., Action Research: Rethinking Lewin. Management Learning 30(2), 127-140 (1999).
16. Oinonen T. Key Performance Indicators for Automated Testing and Continuous Software Integration. Master’s thesis, Aalto University, 70p (2018).
17. Wohlin, C., Runeson. P., Höst, M.,Ohlsson, M. C., Regnell, B.,Wesslén, A. Experimentation in Software Engineering, 236p (2012)