e-Informatica Software Engineering Journal, Volume 12, Issue 1, 2018, pages: 217–236, DOI 10.5277/e-Inf180109

Milestone-Oriented Usage of Key Performance Indicators – An Industrial Case Study

Miroslaw Staron*, Kent Niesel**, Niclas Bauman**

*Computer Science and Engineering, Chalmers / University of Gothenburg

**Volvo Car Group

miroslaw.staron@gu.se, kent.niesel@volvocars.com, niclas.bauman@volvocars.com

Abstract

Background: Key Performance Indicators are a common way of quantitative monitoring of project progress in modern companies. Although they are widely used in practice, there is little evidence on how they are set, and how many of them are used in large product development projects.

Goal: The goal of this paper is to explore how KPIs are used in practice in a large company. In particular, it is explored whether KPIs are used continuously or only during short, predefined periods of time. It is also explored whether software-related KPIs are reported differently from non-software-related KPIs.

Method: A case study of 12 projects at the Volvo Car Group in Sweden was conducted. The data from the project progress reporting on tools was collected and triangulated with data from interviews conducted with experts from the company.

Results: KPIs are reported mostly before the milestones and the manual assessment of their status is equally important as the automated data provision in the KPI reporting system. The trend of reporting software-related KPIs is very similar to the non-software-related KPIs.

Conclusions: Despite the documented good practices of using KPIs for project monitoring, it is difficult to develop a clear status-picture solely using quantitative data from progress reporting tools. It was also shown that the trends in reporting the software-related KPIs are similar to the trends in reporting the non-software related KPIs.

Keywords: software metrics, key performance indicator, project management, case study

1. Introduction

Monitoring large product development projects is a challenging task for project management teams. Project managers, together with sub-project managers at different levels, quality managers and line managers, often use quantitative data to present the progress of projects and the readiness of their products [1]. Key Performance Indicators (KPIs) are used for the purpose of monitoring progress, to capture quantitative data and interpret it [2]. Although this practice is very common and well-known, there are not many studies on how this reporting is done in practice, e.g. how often the KPIs are reported, how many KPIs are used and how quantitative data is used in setting the values of the KPIs. In the literature, the classical use of KPIs is to continuously report and monitor the progress of the development, which usually leads to using KPIs for decision support and dissemination of information about the project status [3]. This study aims at analysing how the literature evidence is aligned with the practice of using KPIs in a large company.

Embedded software development projects require synchronization between software-related and non-software-related sub-projects in order to result in a complete product. Project managers, however, often seek advice on whether the projects should follow a more software-inspired agile way of planning (and thus adjust to the software project management practices) or follow more strict, hardware-inspired project planning and monitoring (thus putting dedicated reporting requirements on software projects). In this paper, the authors explore how large software projects are managed and whether the usage of KPIs is more software agile-like or hardware upfront planning-like.

1.1. Problem statement

The paper explores the problem of understanding the practice of using KPIs. In theory, KPIs should be reported and monitored continuously, in order to provide an up-to-date status of the project. However, in practice, this continuous reporting requires resources and scales poorly with the number of KPIs. It is also the case that, if the number of KPIs grows, the probability increases that multiple KPIs monitor the same or related issues.

The contribution of this paper is analysing both KPIs for software-related KPIs and for the entire project. The state-of-the-art in this area considers either only software development projects, or focuses on project-management aspects. Therefore, it is important to study the KPIs in embedded software projects, where the software development sub-project is contrasted with non-software sub-projects.

1.2. Research objectives and questions

The general research methodology applied in this work is a case study; the methodology which emphasizes close collaboration between industry and academia, and results in changes in hosting organizations. To begin with the following research question is addressed – How are KPIs for monitoring project progress used in practice in a large product development organization? Steyn and Stoker [4] recognized this as an issue, and provided the evidence that different ways of using KPIs impact the performance of development projects. The use of KPIs can also determine whether the company uses the traditional approach to performance monitoring, or it uses the modern principles of Neely et al. [5]. In the research twelve projects were studied focusing on such aspects of reporting as:

- How are KPIs defined? – to understand the structure of KPIs used in industry.
- How often are KPIs reported in practice? – to explore the frequency and thus the cost of reporting KPIs, and to understand how timely the KPI information is provided.
- Who is responsible for reporting and acting upon the definitions of KPIs? – to understand the stakeholders in the process of KPI reporting and decision making.
- How can we statistically identify dependencies between KPIs? – to explore whether KPIs are independent from one another, and therefore to understand whether the number of KPIs is sufficient or too extensive.
- Is there a difference between software-related KPIs and non-software-related KPIs? – to explore whether the software development KPIs are reported differently than the non-software development ones.

1.3. Context

This work studies a large product development organization – the Volvo Car Group, a Swedish vehicle manufacturer. The analysis encompassed 12 car development projects where the number of KPIs varies from 252 to 552 per project. The following definition of a KPI was used – KPI *(Key Performance Indicator)* is a customizable business metric utilized to visualize the status and trends in an organization. A KPI has an owner (a stakeholder according to ISO/IEC 15939 [6]), an interpretation (an analysis model according to ISO/IEC 15939) and is linked to a business strategy of the organization. This definition of a KPI is consistent with the use of the term in well-established methodologies, such as the Balance Scorecard [7].

The remaining of the paper is structured as follows. Section 2 presents the theoretical framing of work. Section 3 describes the design of the case study. Section 4 presents the results and answers to the research questions. Section 5 discusses the results in the light of the existing body of knowledge. Section 6 summarizes the paper and presents conclusions.

2. Background

In order to study the use of KPIs in the company the authors used a set of models showing how the reporting process is done. The models are presented in Figure 1. The models are divided into four groups of activities:

- storage: the way in which information, needed to calculate the KPI, is stored – it could be either a database, such as a product article database or personal assessment of, e.g., whether the quality of a requirement is sufficient,
- extraction: the way in which the information is provided to KPI systems – it could be manual reporting or the automatic extraction of information using a script (for example by counting the number of defects reported in a database),
- analysis: the set of methods for analysing the values of KPI, they assess the status (set the colour of a KPI – green, yellow or red) – it could be an algorithm, using a set of pre-defined criteria, or a manual assessment,
- presentation: grouping activities related to the presentation of the material, which can also be either manual or automated – the manual presentation can be in the form of an MS PowerPoint presentation and the automated one can be a web-based dashboard with indicators [8].

These four groups of activities are based on the measurement information model defined in the ISO/IEC 15939 standard (Software and Systems Engineering – Measurement Processes), [6].

The set of models used for the theoretical framing of the KPI usage, comprises four models which have distinct characteristics.

The most basic model is manual reporting, the analysis and presentation of the KPI values, as initially presented by Kapplan and Norton, as part of the Balanced Scorecard methodology [9]. In this model (M), the focus of KPI usage is on the periodical reporting and monitoring of organizational performance. The data to calculate the values of KPIs is often available through individuals (e.g. by filling in reports) and needs manual assessment. The KPIs, in the M model, are often updated periodically and are prone to missing data points, however, it is very flexible. This model can be observed in the studies discussion early adoption of the Balanced Scorecard [9].

The next model, which is more advanced, is the M-A model, where the extraction activities are automated, but the assessment of the KPIs status is manual. This kind of model is prescribed by many project management tools and methodologies which focus on the quantitative assessment of project progress and performance. An example of such a method is PRINCE2 [10]. The M-A model can be observed in modern companies utilizing business intelligence tools. The KPIs in the M-A model are updated continuously and are analysed periodically; in practice this means the same disadvantages as the M model with a reduction of problems coming from the missing data points. The M-A model can be exemplified by such cases as surveys for customer satisfaction [11].

The next model is the A-M model, where most of the data extraction and presentation tasks are automated. The assessment of the values is, however, still manual. An example of such an assessment is the quality of the product under development by counting the number of defects discovered during testing. As the automation of the extraction and presentation is used, the KPIs in this model are often used as measures, and visualized as trends – since they are collected continuously. However, their assessment is periodical, which means that the status is available at certain points of time. Thus KPIs in this model are more difficult to interpret, but easier to visualize [12]. An example of this kind of a model is the set of KPIs at Volvo Car Group where the data collection is automated but the setting of colour is manual [13].

Finally, the most advanced model is the A model, where all tasks are automated. The stakeholders of the KPI pre-define rules for analysis and these rules are applied automatically for the measures collected. This kind of model has been shown to be an efficient way of collecting the information and supporting decisions [14,15], [16]. KPIs which are used in this way require maintenance (evolving criteria, updating data extraction programs), but require no manual effort on a daily basis. It is the automation that makes it very attractive for modern companies. The main disadvantage of this model is, however, the fact that not everything can be calculated automatically, which in practice leads to the degradation of the A-M or M-A models. The main advantage, on the other hand, is the ability to provide status assessment continuously; thus enabling the development of information radiators or dashboards spreading information across the project team. Examples of this type of reporting can be found at such companies as Ericsson.

The set of models used in this study as the theoretical framework, allowed to clearly identify the patterns KPI use. The way of updating KPIs in these models allowed also to identify the possibility of visualizing the status in the long run – potential for the development of dashboards.

3. Case study design

This section describes the design of the study, following the guidelines by Runeson et al. [17].

3.1. Research questions

The following research question were addressed – *How are KPIs for monitoring project progress used in practice in a large product development organization?* In order to address the question, a number of organizations within the Volvo Car Group were studied – they were involved in product development, manufacturing engineering, provisioning of parts for production and contract management. The study encompassed twelve projects and focused on such aspects of reporting as:

- How are KPIs defined? – to understand the structure of KPIs used in industry.
- How often are KPIs reported in practice? – to explore the frequency and thus the cost of reporting KPIs, and to understand how timely the KPI information is provided.
- Who is responsible for reporting and acting upon the definitions of KPIs? – to understand the stakeholders in the process of KPI reporting and decision making.
- How can we statistically identify dependencies between KPIs? – to explore whether KPIs are independent from one another, and therefore to understand whether the number of KPIs is sufficient or too extensive.
- Is there a difference between software-related KPIs and non-software-related KPIs? – to explore whether the software development KPIs are reported differently than the non-software development ones.

Exploring these questions, provides a possibility to understand whether there is a minimum viable set of KPIs to be used in a project, and how to construct a dashboard for visualizing the status of the development in an automated way. For example, in order to construct a real-time dashboard as advocated by Azvine et al. in the context of telecommunication industry [18]. Understanding whether the status of a KPI (or its colour) is usually set using quantitative data from source systems, provides us with a possibility to automate the process of setting the KPI status and thus decrease the cost of project monitoring without a decrease in its quality.

Understanding how KPIs are used in practice requires a combination of analyses of data from different sources and, therefore, two different sources of data collection were triangulated – documents at the company (in the form of a project status reporting tool) and interviews with stakeholders who report on the progress.

3.2. Case and subject selection

The study presented in this paper was conducted over a period of six months. The research was done based on interviews with stakeholders at multiple units of the company – Electrical System and Electrical Propulsion, Powertrain, Chassis, Purchasing and Manufacturing Engineering. The interviews are complemented with the statistical analyses of historical KPI change data from finished and ongoing projects. The statistical analyses are done based on constructing

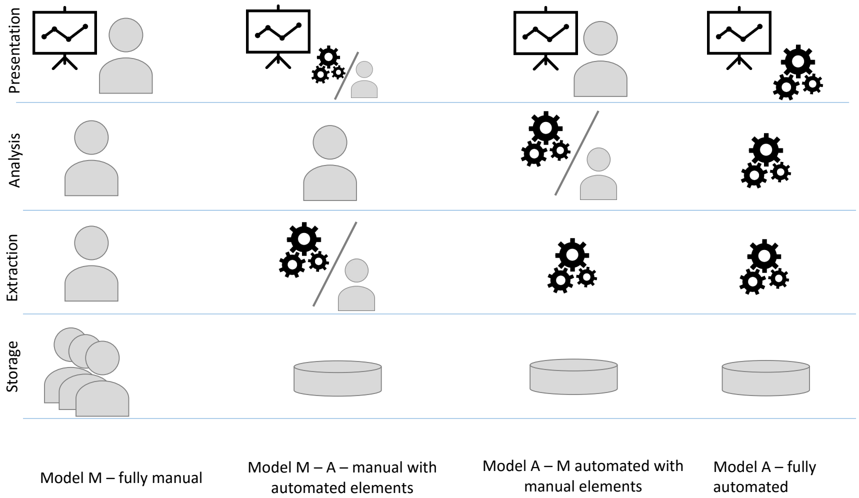

Figure 1. KPI reporting models

co-dependencies, co-change models of KPIs for the selected set of 12 projects.

The projects are selected based on their characteristics – from minor year-model update projects to large complete new vehicle platform projects. The aim was to cover a variety of projects. Table 1 summarizes the characteristics of each project.

The minor model update projects are usually limited to the updates of vehicle parts in this study called disciplines[1] (e.g. new infotainment software, engine control software update), and therefore are limited in scope, but not in the process as all process steps need to be conducted. Studying the smaller projects helps to establish the minimum number of KPIs , whereas studying the large projects allowed to understand how many KPIs are required in large projects. In the table, the number of KPIs as a proxy for the size of the project is used, because in these analyses defined the number of KPIs to correlate with such project parameters as project length, effort and cost.

The number of KPIs reflects the size of the project as larger projects tend to require more monitoring and control than smaller ones. Therefore, project B is the largest one, both in terms of the number of KPIs and the size (confirmed through interviews). Project A is in the middle of the size spectrum (386 KPIs in the set of 252–552 KPIs) and was chosen as a good pilot project to study in detail; the data from this project was used as examples in Section 4. The industrial partners in the research provided this information and also showed evidence for that. However, these numbers cannot be reported outside the company.

For the interviews, the subjects were selected based on their experience. All subjects had had over 10 years of experience in project management and also in managing software and car development projects at the studied company. Each of the subjects was recommended by his or her manager as the most knowledgeable person in that area. All subjects were involved in the subset of the projects studied, although not all respondents were involved in all projects.

3.3. Data collection procedures

This research study was divided into four distinct parts, as presented in Figure 2 – statistical analysis of the co-changes of KPIs and interviews about using KPIs and future needs.

Figure 3 presents an overview of the research process for the statistical KPI co-change analysis. The process comprises four steps – starting from the exporting of the KPI change data from the database and finishing with the prioritization of

[1]Disciplines are the parts of development, e.g. active safety systems development, engine development, powertrain electronic control unit development, new production line development.

Table 1. Main characteristics of the projects studied

Project	KPIs	Characteristics
Project A (pilot)	386	Minor year model update of a mature car model. Development done on a single site, includes all disciplines.
Project B	552	New platform development project. Development done on a single site, includes all disciplines.
Project C	396	New functionality development for an existing platform. Single site, including a subset of disciplines.
Project D	382	New functionality development project. Development done on a single site, includes few disciplines only.
Project E	257	New functionality development project. Development done on a single site, includes few disciplines only.
Project F	442	New engine development project. Development done on a single site, includes few disciplines only.
Project G	305	New functionality development project. Development done on a single site, includes a subset of disciplines only.
Project H	252	New functionality development project. Development done on a single site, includes few disciplines only.
Project I	421	New functionality development project. Development done on a single site, includes few disciplines only.
Project J	342	New functionality development project. Development done on multiple sites, includes all disciplines.
Project K	494	New functionality development project. Development done on a single site (different than projects A–J), includes few disciplines only.
Project L	431	New engine development project. Similar to project F, but on a different engine. Development done on a single site, includes few disciplines only.

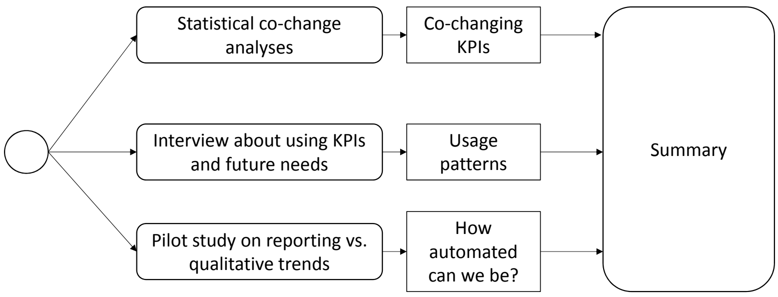

Figure 2. Overview of the research design

candidate KPIs to remove. As shown in the figure, the export of the data from the database in step 1 results in one KPI change per project, and the analyses in step 1 and step 2 are conducted per project. The analyses in step 3 and step 4 are done for all projects, by consolidating the results from each individual project.

In step 1, the exports result in text files (.csv) with the change analyses in the format: <kpi name, value, change date>. Grouping these changes results in the statistics of how often each KPI changed together with another KPI.

The interviews were conducted with 12 different respondents – project managers (1 person), sub-project managers (6), unit project managers (4) and quality responsible (1 person). Each of these people represented a different department at the Volvo Car Group, and each worked with a number of different projects (including the set of projects in our sample). They represented departments responsible for powertrain development, purchasing, quality management, electrical system development (including software) and interior development. The following questions were asked:

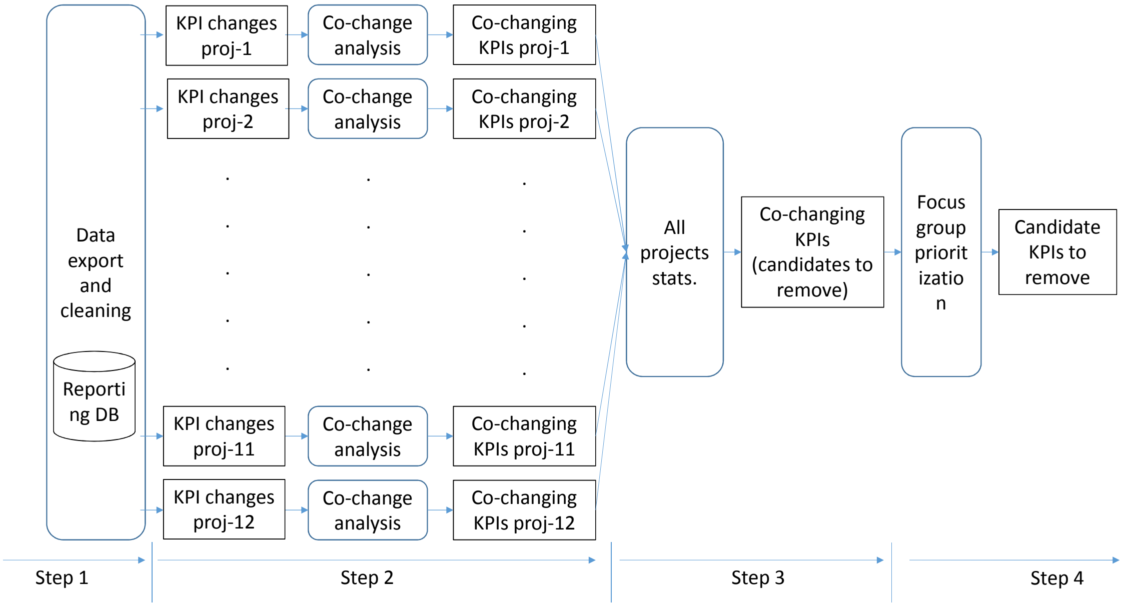

Figure 3. Overview of the statistical analyses

- Which elements of the construction are you responsible for?
- What is the main focus of your work with KPIs?
- How do you update KPIs today? – daily, before the milestone or another frequency?
- When do you discuss the status of the indicators and with whom?
- Which of your indicators are "automatically" imported from other systems?

For each question, the answers were discussed. We also asked about the dependencies between KPIs, their definitions, asked for an exemplification of how the stakeholders work with the persons who define the KPIs.

3.4. Analysis procedures

The first analysis of the patterns of changing KPIs was done by visualizing the changes using a heatmap [19], [20], which is a graphical representation of contingency tables. The heatmaps allow us to:

- identify KPIs which frequently change in the project,
- identify which KPIs change only at a particular point of time, and
- check if progress reporting is done continuously or periodically.

A he change in a KPI was defined as an event when a stakeholder actively updated the status of the KPI, which means that there is an update event in the database where the KPI status was updated. Process-wise, this corresponds to the situation when a stakeholder needs to make an active assessment of the value of the KPI. He/she sees a notification on his dashboard and should make an active choice (even if it is only to confirm that the status is the same (e.g. still "green"). This means that the project manager at a higher level has confidence that KPIs status is up-to-date.

A co-change was defined as the update of two KPIs in the same period of time – in this case during one day.

These patterns are used for further analyses of co-changes and quantifying these changes as percentage. The dependencies are identified using the method developed in the previous research of the authors to monitor co-dependencies in software modules [21], using the co-change model presented in Figure 4.

The figure presents the lifeline of one assignment (e.g. a project) with the changes in KPIs. Two KPIs are considered as potentially dependent on one another if they change within the same day in the majority of days. For example, if KPI-A changes 10 times during the period

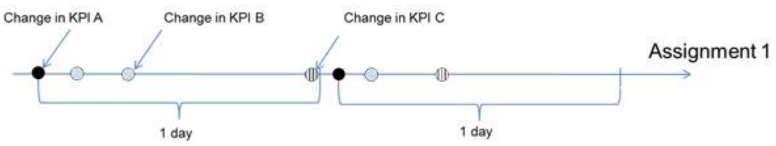

Figure 4. KPI dependency analysis model

of 10 days (once a day), and KPI-B changes 8 times during the period of the same 10 days, then these two KPIs can be considered as 80% dependent. This kind of analysis allowed to identify superfluous KPIs and to understand potential dependencies between them.

After calculating these dependencies in a single project, the dependencies were summarized for all studied projects. In this way, dependency pairs were obtained for all projects. In order to sort the changing KPIs, the PRE(x) function, which is defined as a percentage of the projects which have a co-dependency of strength x or more, was used. However, there could be cases when a KPI is available only in one or two projects. In such a case the AV(x) function was used as another criterion. The AV function is defined as the percentage of the projects where a KPI is used. For the analysis purposes PRE(75) and AV with the cut-off point of 33% (dvs. KPIs present in at least 33% of the projects) were used.

The data from the interviews was analysed by the main author of this study using coding. The results were discussed with the reference group for the project consisting of the other co-authors, two line managers and two experts at the company.

4. Results

This section presents study findings structured by research question.

4.1. How are KPIs defined?

The definition of KPIs consists of two parts – the measurement method describing how to collect the data for a given KPI and the decision criteria describing how to set the colour of the KPI (red, yellow or green). This means that the definition corresponds to the groups of activities prescribed by the theoretical framework adopted in this study (Section 2 – extraction and analysis). The KPIs are visualized using a web portal, as presented in Figure 5.

It was found that the measurement method for the KPIs could be defined in two ways:

– measuring that an activity has been performed (digital answer yes or no) – for example that a review of requirements has been performed, or
– counting the number of elements of a given type – for example how many defects of a specific type were discovered

These two measurement methods correspond to two models M-A and A-M. Both include the evaluation source systems before reporting a KPI – one is done automatically by extracting information (counting the number of elements) and the other one is the measurement that an activity has been performed (digital answer).

It was have found that these two measurement methods (and thus the reporting models) are used interchangeably, but their frequency changes over time. In the early stages of the project, it is common to use KPIs measuring the performance of an activity. In the late stages of the project it is more common to use KPIs representing the count of product elements – how ready the project is for release.

During the interviews, it was found that this transition from the reporting of activity progress to the reporting of product readiness is common throughout the project. Counting a number of elements is used for product-related KPIs, as it is easier to count elements that are "ready" or "tested" towards the end of the project (where the product becomes more tangible for the project). The process related KPIs showed that an activity was performed and therefore they were more common towards the beginning of the project.

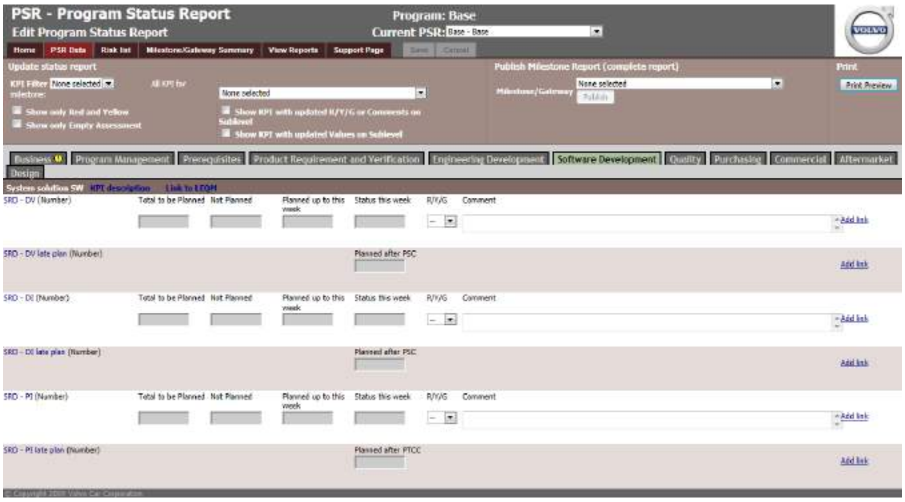

Figure 5. Web interface for status reporting

An example of the process-related KPI, which is calculated by measuring whether an activity has been performed, is the Requirement reviewed KPI for one of the early milestones in the project. The KPI has a colour set to green when all requirements are reviewed, to yellow when not all requirements are reviewed but there is a plan how to review all requirements and red when not all requirements are reviewed but there is no plan how to achieve them.

An example of the product-related KPI is the software product quality KPI. The KPI is calculated by counting defects which have a certain severity. The rules for setting the colours of the indicator depend on the phase of the project. Although it is calculated only in the last milestones of the project, the criteria for setting the values include the severity of defects and the number of defects. A criterion for setting red at one of the milestones for this indicator is:

– Green: number of defects with Severity 1, 2, 3, and 4 is 0 in status New or Open.
– Yellow: not meeting the target but with agreed plan in place to achieve target.
– Red: number of defects with Severity 1, 2, 3, and 4 is more than 0 in status New or Open or any (Severity 1, 2, 3, and 4) Passed Requested Target series.

The second type of KPIs – based on counting the number of the elements of a specific kind – are more quantitative in nature and that criteria for setting the colours (levels) of the KPI are clearer than for the first type of KPIs – based on measuring that an activity has been performed.

The interviews allowed to establish that the KPIs of these two kinds are mixed and that there is a need for more alignment. The interviewees also mentioned that having both types of the KPIs makes it difficult to visualize the status of the project at a specific moment – as some of the "greens" never change (performance of an activity) and some of the "greens" might change over time (number of defects).

4.1.1. An example of a product-related KPI – Software Product Quality

This KPI uses a defect tracking database as the source system. The summary of both the number of KPI updates per week (the top chart in the figure) and the data in the source system (the bottom chart in the figure), is presented in one diagram in Figure 6.

The colours of the bars in the top chart indicate the colour of the KPI reported. The colours of the bars in the bottom chart in the figure indicate the different status of the open defects in the database. The lines in the bottom chart show the cumulative number of defects reported in the entire project.

The criteria for setting KPI colours are related to the timing and the number of open

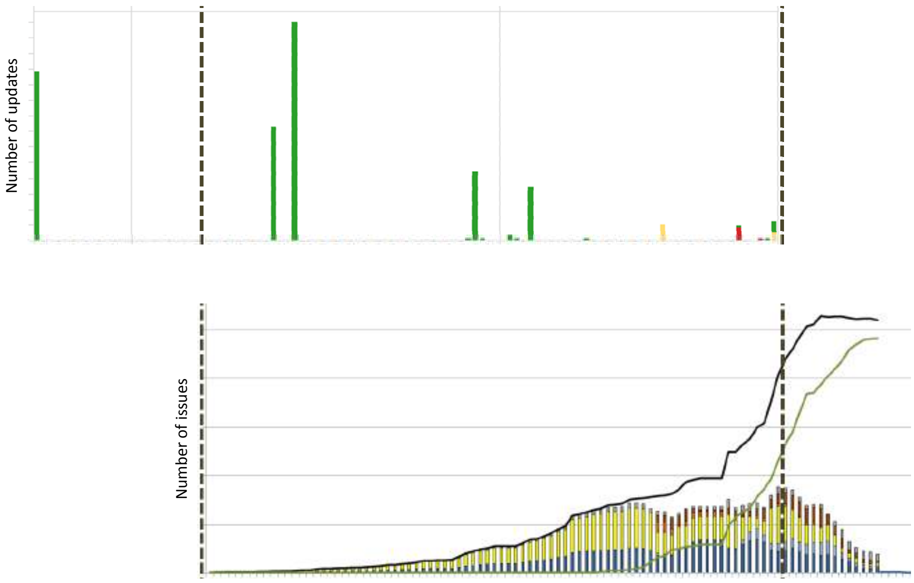

Figure 6. Comparison of updates of KPI and the underlying data from source system (defect reporting database). The vertical lines show alignment in time. Due to the sensitivity of the data the scales have been removed

(non-resolved) defects. For example, in order to set the status to yellow, there should be at least one newly reported defect after a specific milestone in the project. In order to set the status to red, the newly reported defects have to come after yet another (later) milestone. This is the case in this pilot project and our interviews have confirmed that the colours of the KPIs for this indicator are indeed set based on these criteria.

Since these criteria are so well defined, in terms of measurable quantities (number of defects in a specific status and milestone), KPI reporting could be supported by the pre-setting of the status of the KPI; thus, reducing the burden of searching for data for the sub-project managers.

4.2. How often are the KPIs reported in practice?

In order to study the patterns of KPI changes per week, a contingency table which summarized the number of KPI changes per week was calculated. Their visualisation was made using heatmaps, as it is shown in Figure 7. As the figure shows there are visible "vertical" lines where a set of KPIs changes the status. This indicates that the project management focuses on "milestones" when reporting KPIs to the database. This finding was also confirmed in the interviews. The reasons for this can be the lack of use of the KPIs between the milestones, and the need to prioritize other assignments.

During the interviews, it was also found that, given the non-continuous update of KPIs, it is difficult to obtain the overview of the current status. If a vertical line is drawn in the figure – a snapshot, it is not clear how "old" the status of each of the KPIs is.

To summarize the data, a histogram of the percentage of KPIs that change over one week was used, as presented in Figure 8.

Figure 9 shows the frequencies of KPI changes per week. Each row represents a project and each column represents the percentage of KPIs that changed per week. Each bar represents the number of weeks when a given percentage of KPIs changed.

As shown in the figure, there is no project where over 60% of KPIs changed and in the majority of weeks less than 10% of KPIs were changed. This shows that model A is not applied at the company as it is characterized by the continuous update of KPIs. During the in-

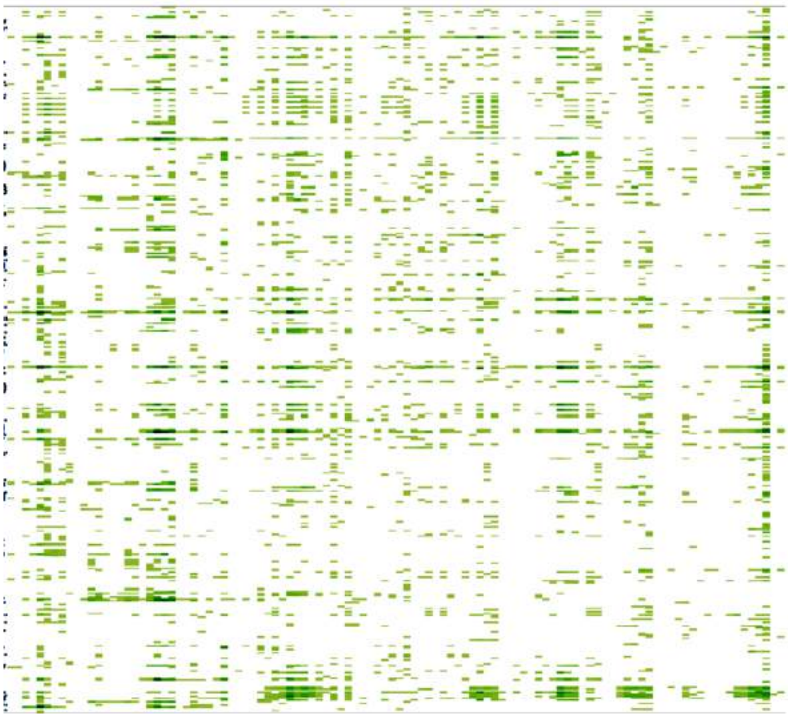

Figure 7. Frequency of reporting KPIs per week. Each row is one KPI and each column is one week; the intensity of the color indicates how often the KPI was updated during given week

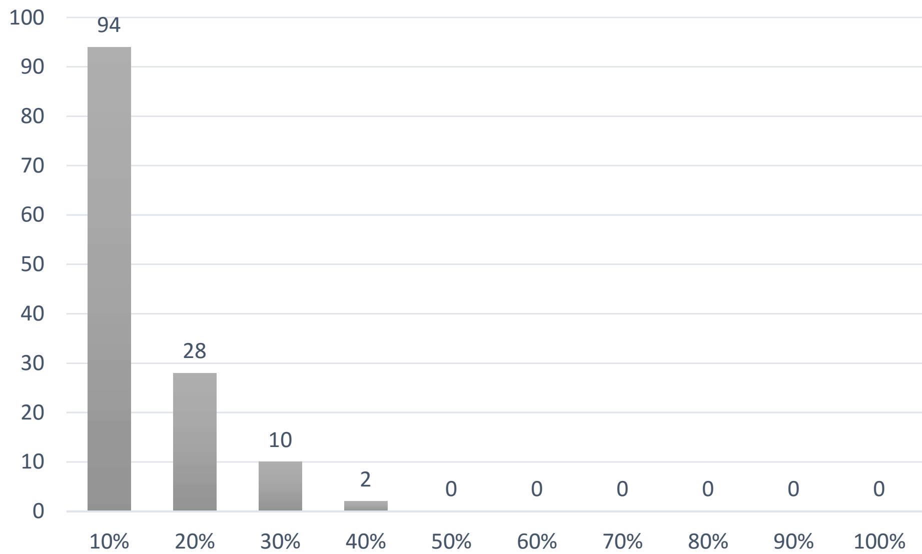

Figure 8. Histogram over the frequency of changes per week – each bar represents the number of weeks when the given percentage of KPIs changed

terviews it was found that model M is not applied either, as KPIs are calculated based on the data from source systems (e.g. requirements database, project planning tools). The interviewees explained that, depending on the indicator, they apply either model M-A or A-M.

At the beginning of the project, it often occurs that model M-A dominates, as the KPIs used at the beginning focus on tracking activities; whereas towards the end of the project it is model A-M which dominates, as the product data is often used for KPI calculation.

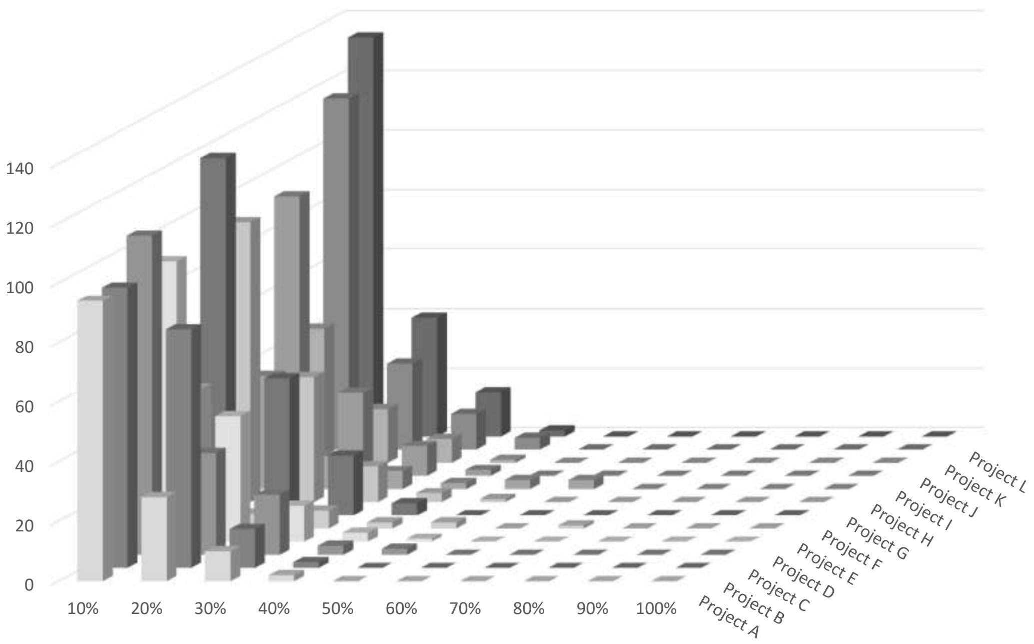

Figure 9. Frequencies of KPI changes in all studied projects

The interviews showed that this milestone-orientation was common – all interviewees supported that. They also indicated that one always needs to comment on the "red" status and therefore the list of risks is an important complementary tool. Each risk can be linked to a KPI and the risks are discussed during project status meetings.

The interviewees showed that, in practice, model A-M was challenging as it was difficult to assess the status of continuously changing data. In the cases of continuously changing data, one needs a fully automated model A to keep the pace of status assessment.

Another challenge identified, which requires the automated model A, is the fact that data import from source systems can be out-of-sync due to the large complexity of the source systems. Interdependencies between systems make it difficult to import all the data at once, and therefore the imports have varying frequency – which makes it difficult to make manual assessments (one does not know exactly how "fresh" the numbers are, i.e. has the data of low timeliness).

4.3. Who is responsible for reporting and acting upon KPI definitions?

The interviews showed that KPIs are reported by project managers and their sub-managers. However, they are always approved by the line managers or the project management team.

The sub-project and unit project managers are responsible for the assessment of KPIs (setting the colour based on the criteria) and for presenting them for the main project manager. When they make the assessment of a KPI, they present their assessment to the line managers of their respective unit or group. The manager approves or adjusts KPI assessment. It was found that it was formally the manager (in all cases), who was responsible for a given KPI.

After the approval of a KPI by the manager, it is the sub-project or unit project manager (depending on the level), who presents the KPI to the subsequent higher level. Sub-project managers present KPIs to unit project managers and the unit project managers present KPIs to the main project manager.

The total project status is presented by the main project manager to a project steering group. The project steering group is the outmost responsible body for the project and can make such decisions as increasing the funding of the project, if needed.

4.4. Which KPIs are co-dependent on one another?

In this analysis the co-change analysis was used. When presenting the results, however, KPIs are listed based on two parameters according to the research methodology described in Section 3:

- PRE(x) function which is defined as a percentage of projects in which the co-dependency of strength x or more occurs, and
- AV function which is defined as a percentage of projects where a KPI is used

Together with the company stakeholders, the authors identified the thresholds for these two functions to be: PRE(75) = 100% (meaning that in all projects where the KPI pair is present the strength of dependency is 75%) and AV = 33% (meaning that a KPI pair is present in at least 33% of the studied projects, i.e. 4 projects). The application of these functions to the data set (over 244 000 pairs of KPIs) showed that there were 75 pairs that were manually reviewed by the stakeholders to establish if they overlapped. Eight (8) pairs were found to be dependent on each other, 27 pairs were assigned for further investigation to test if the statistical dependency can be confirmed by experts, 3 pairs were already removed (after the end of the projects and before the end of the study) and 37 pairs were found to be false positives. The 37 KPIs which were found to be false positives were present in only 4 projects, whereas the eight pairs found to be co-dependent were found in all 12 studied projects. The 27 pairs assigned for further investigation were found in between 5 and 9 projects from the 12 projects sample.

The interviews revealed that the majority of the KPIs in the list of 75 KPIs were progress indicators. These progress indicators were used to indicate that sets of activities were successfully conducted based on the set of pre-defined quality criteria.

During the interviews, it was found that the KPIs which are related to a specific deadline were usually reported in the period of four weeks before the milestone until one week after the milestone. The four weeks period allows the project (and subproject/unit project managers) to focus on the goal and report the KPIs which are important for assessing the completion of the milestone.

4.5. Is there a difference between software-related KPIs and non-software-related KPIs?

This analysis, allowed to identify which KPIs were software-related, apart from this a correlation analysis between the software-related and non-software-related KPIs was conducted. The correlated elements were the trends in the reporting of these KPIs. As a result, 20 KPIs to be related to software development activities were identified. Once these software-related KPIs were identified, they were added to the changes so as to create a time series for these changes. Then the same analysis was performed for the other, non-software-related KPIs. An example of this chart is presented in Figure 10.

The diagram indicates that the changes in the KPIs per week follow the same trend. In particular, the visual analysis shows that the peaks in the number of KPIs reported happen at the same time. The Pearson correlation coefficient for these two series is 0.69, which is a strong correlation.

It can be observed that the peaks in the number of KPI changes is similar. However, it can also be observed that the software-related KPIs have more peaks. This can be explained by the fact that the software-related KPIs have higher similarity to each other than non-software-related KPIs. The non-software-related KPIs contain

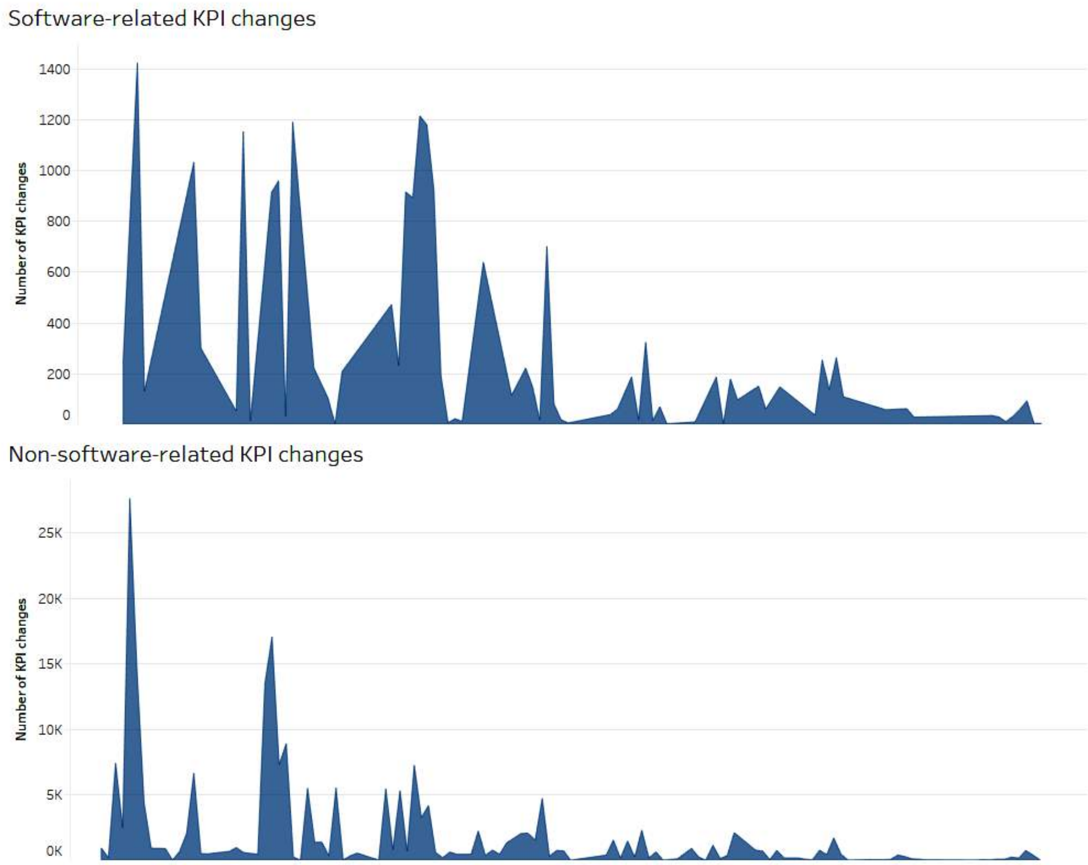

Figure 10. Changes in software-related and non-software-related KPIs per week

all other disciplines (except for software), which explains the lower similarity in that group.

Table 2 presents the results of the correlation analysis for all studied projects.

Table 2. Pearson correlation coefficients for time series of number of changes in KPI values for software-related vs. non-software-related KPIs

Project	KPIs	Correlation
Project A (pilot)	386	0.69
Project B	552	0.69
Project C	396	0.65
Project D	382	0.73
Project E	257	0.86
Project F	442	0.83
Project G	305	0.73
Project H	252	0.72
Project I	421	0.61
Project J	342	0.63
Project K	494	0.84
Project L	431	0.86

The table shows high correlations, which means that the trends are similar. This in turn means that the reporting of software-related KPIs is similar to the non-software-related KPIs.

4.6. Summary – How are KPIs for monitoring project progress used in practice in a large product development organization?

Based on the results, the following trends in KPI use were observed:

1. The definition is standardized using three levels – red, yellow, green.
2. The trend in reporting software-related KPIs is similar to the trend in reporting non-software-related KPIs.
3. The definition of each of the levels is based on the ability to act upon the problems indicated by the KPI – if the status is yellow, then the

responsible project manager has a plan how to get back on track before the next milestone; if the level is red, then the project manager registers the risks in the qualitative description of risks.

4. There is some progress from process-oriented to product-oriented KPIs over the project time – at the beginning of the project, when no product is ready yet, the control of KPIs whose activities have been performed (e.g. requirement reviews are completed). At the end of the project, when there is a product, KPIs monitor if the product is ready for release (e.g. that all ECUs are tested to 100%).
5. The reporting of KPIs is done based on milestones – unlike the standard software development projects, KPIs are used to check whether a project has achieved the degree of readiness required to move to another phase (e.g. whether all software components for an ECU are ready to be tested in a simulated environment).
6. The responsibility for the development of a KPI is given to measurement champions who have the necessary knowledge about the domain and the specifics of the product.
7. The responsibility for the reporting of KPI value is given to the project managers at different levels as they are responsible for ensuring that the product and project follow the set resources.
8. The responsibility for the formal approval of the KPI status is given to the line managers of relevant organizations, as they have the responsibility for resources in a company.

The above items show that project management in the automotive sector, including embedded software projects, follows the classical, gate- and milestone-oriented principles. According to these principles, planning and following the plan are central.

The fact that the number of KPIs is between 252 and 552 shows that the complexity of the development is too high to change to the continuous reporting of KPIs. Therefore, there is a need to combine manual assessment with automated data provisioning from source systems.

4.7. Validity analysis

In this paper the framework of Wohlin et al. [22] and a more recent framework by Runeson et al. [17] were followed in evaluating the validity of the research.

The main threat to the external validity is the fact that only one company is studied – Volvo Car Group. Single-case studies can risk bias towards informants or the context. However, the authors believe that a broad sample of projects helps to increase the external validity at least beyond one project. This study is also considered an important contribution to the studies of software development as part of a larger context – car development.

The main threat to the construct validity is the potential mono-operation bias as the study was conducted at a single company only. In order to minimize this threat by diversifying the project sample – the authors chose a sample of 12 projects with different characteristics, scopes and project teams. Another threat which was minimized is the mono-method bias – using a single measure of co-dependency. It was minimized by using the same measure as in the previous studies where the validity of finding co-dependencies was established (although for a different type of a measured entity). Finally, another identified threat related to the measurement of co-change was the fact that some co-changes are purely random. To minimize the threat, the PRE(x) and AV(x) functions were used to set thresholds to reduce the probability of capturing a random change as a valid co-change. It was also taken into account that the stakeholders do not update the status of a KPI when the value changes and the status remains the same (e.g. "red"). This is a threat that could not be minimized as there was no measurement method to capture this situation, however, it was established that this situation seldom occurred in the studied company.

Using the theoretical model, shown in Figure 1, the study was constructed in such a way that each answer was coded based on the elements of the model. This introduced the risk of missing important information, which was limited when using alternative methods (e.g. grounded theory). However, the model was chosen as interviews were to be used as a triangulation method to the quantitative data analysis.

In this study the risks related to the internal validity and reliability were to be minimised. Therefore, the triangulation of data collection methods, i.e. statistical and archival analyses (KPI data from the database) and interviews, was used. The triangulation allowed to check the root causes of the patterns visible in the statistical analyses – e.g. identifying potential causes of KPIs being co-dependent.

In general the main threat to the conclusion validity of such studies as the one presented in this paper – identifying co-dependencies in numerical data sets – is drawing conclusions based only on statistics. Thus, it was decided to triangulate the data sources (document analysis and interviews) and these triangulated interviewees and their roles (multiple organizations within Volvo Car Group and multiple roles) were used. The triangulation also minimized the risk that the results would be biased or would represent a specific part of the organization.

Since interviews were used in the study, the authors are aware of the conclusion validity threats related to overinterpretation. In order to minimize these threats, the findings were confirmed during a workshop with all the interviewees and in another workshop with a reference group for the project (consisting of technical experts and managers).

5. Related work

In a recent study Todorovic et al. [23] explored the types of KPIs which can be found in organizations focused on projects. They found, which was confirmed in this study, that there are such types of KPIs which are related to progress and such that are related to performance and that the latter are more challenging. It can be thus concluded that the theoretical models, published in software engineering literature about KPIs, need extensions to improve the formulas by adding uncertainty components or temporal aspects, or both. In a recent study, Todorovic et al. [23] identified the properties of KPIs which are important for this study, too – a KPI being actionable and measureable. In their study Todorovic also postulated the fact that there is a difference between measures and KPIs which corresponds to a similar difference between progress KPIs and performance KPIs. The latter are naturally important, but as Todorovic et al. state, also very challenging because in the project oriented organizations the progress KPIs tend to be the most common ones. However, even though it could be observed that in practice the theoretical framework of Todorovic et al. can be applied, their models, however, cannot because the models are based on the assumptions that there is a known and updated status at the moment when the model is applied.

Pilgoret [24] lists a number of KPIs which are important for modern project managers. The list includes both process and performance KPIs, but does not show when and how the KPIs should be reported. In this study new evidence showing that not all KPIs are equal and that they are reported only when necessary is provided. The evidence that KPIs are not reported continuously and at the same time they are reported more than once per project means that there is a change in the way in which KPI are traditionally used. An example of such a study is the study of Pilorget [24] who shows the estimations of a number of KPIs (although much smaller than the number found in our study) used only once during the project. Since the KPIs change and get updated, methodologies like this need to be adjusted to support more continuous re-evaluation of the project status value and should include the temporal component – how “old” or “stable” the data in the KPIs is.

Colin and Vanhoucke [25] presented a recent study on statistical performance control approaches for project monitoring. The results recognized the challenges with continuous monitoring of project status with KPIs, which is addressed in this work.

The use of KPIs is very common in the field of corporate performance measurement and the predictions are that it will become even more important[3]. In the interview with the main experts behind the Balanced Scorecard approach, this trend becomes even more evident. Therefore, in this study we intended to explore the use of KPIs in practice, and explore what the industrial trends in this area are.

In a study by Jaafari [26], the challenges with milestone-based project reporting were recognized. The evaluation of the presented PH-Check approach in the industrial context showed that the milestone-based assessment leads to efficient identification of shortcomings, and provide the ability for stakeholders to react. This positive view of the milestone-based reporting is shared by some of the interviewees in the presented study.

As the modern product development evolves, so do the practices related to it. In a recent study of how the golden triangle in project management is perceived on the lowest levels of the organization, Drury-Grogan [27] found that quantitative information is of less value for lower levels. Instead, the consensus and understanding of project goals become more important as the responsibilities of development teams increase. For cross-functional software development teams, as Drury-Grogan found, the product is of more focus than the project. This study also aimed at understandnding whether these findings were valid for car development projects, too.

The rationale behind this study was to follow up on the study of Steyn and Stoker [4], who found that there was a significant difference between the performance of projects, depending on which project measurement methodology was chosen. They found that measuring such parameters as contingency task allowance can increase project performance. Although limited to a small number of measures, the study shed light on the use of measurement methodology which can make a difference in projects. Therefore, it was decided to study the practice of how the measures (in this case KPIs) are used and when they are reported. In the light of the research of Steyn and Stoker [4], it could be concluded that it is perhaps not the use of measures as such but the use of measures related to organizational goals (in the same sense as Todorovic et al.'s study) can make a difference. It was found that performance oriented KPIs stimulate goal orientation and can be (in principle) updated continuously which increases the probability of project success. This means that in project management measurement the separation of these two concerns – progress and performance oriented KPIs reporting should be postulated.

Raymond and Bergeron [28] surveyed 39 project managers with regards to the effects of the use of reporting systems in project management. The results showed that the use of project management information systems improved the efficiency and effectiveness of managerial tasks in projects. These results were important as an input to discussions with the industrial partners in this study, and were confirmed in this research – these kinds of information systems provide value. However, it was found that it is important to find the right balance between the cost of reporting and the value obtained.

Marques et al. [29] presented a method for aggregating the status of KPIs (and metrics) in complex products. Their case study illustrated how feasible aggregation is in practice. Marques et al.'s approach is an alternative to the use of multiple KPIs at the total project level, which is the case in the studied company.

In our previous work, the use of KPIs was studied in software development organizations, exemplified by Ericsson [30], [31]. The findings showed that the number of KPIs can be small and that the automation is the key aspect. The results from the study, presented in this paper, contradict the results from the previous studies – lower degree of automation and larger number of indicators. This can be explained by the fact that the previous studies investigated software development projects, whereas this study investigated both software and non-software development projects, as car development projects combine significantly more disciplines than software development projects.

Sanchez and Robert [32] provided a framework for defining KPIs, where they combine standard KPIs, such as the earned-value indicator with indicators related to risk management. As a result of the analysis of the pattern of reporting risks in the studied projects, this case study provides evidence that this kind of combination is very much needed, as the risk-view of the KPI provides a more complete view on project performance.

Choosing a set of metrics and KPIs for monitoring *f* projects was also studied in the Netherlands [33], with results showing that the most important determinant of the success of KPIs adoption is the behaviour of the organization. The studied organizations identified a gap between the research and development function and the R&D department's responsibility, which can be observed in the studied projects - the discrepancy between the process and project.

The quality of KPIs was also studied in a number of contexts, and, as a result, an interesting study was by Spangenberg and Göhlich [34], who studied how to construct the roadmaps of mechatronic software systems using KPIs. In their context of transportation systems, technology readiness levels were combined with goal-oriented KPIs, such as carbon dioxide emissions, in order to allow the simulation of project outcomes. The presented study provides evidence that this kind of goal needs a change in the mindset of project management, as KPIs are progress oriented at the beginning of the project. However, this kind of simulations are theoretically possible towards the end of the project, where the focus shifts towards product-oriented KPIs.

Lainhart et al. described the methodology for software project governance – COBIT, [35] – which provides a holistic approach to managing software projects. Our work provides evidence that milestone-oriented project monitoring (as described in the COBIT processes part) is an important industrial practice.

Finally, this work can be considered an input to planning measurement programs using frameworks, such as GQM+Strategies, [36], [37]. They help to establish and evolve measurement programs by defining KPIs and relating them to company strategies. Our findings, that KPIs are used when assessing milestones, can be an input to defining KPIs which are to be used in the way familiar to the automotive software engineering industry.

6. Conclusions and further work

This study explored the question of how KPIs for monitoring project progress are used in practice in a large product development organization. The study encompassed the analysis of how KPIs are used in practice by investigating 12 car development projects at the Volvo Car Group. Interviews and statistical co-dependency analyses were used to explore how many KPIs exist and how often they are reported. The reference for the analysis were four theoretical models.

The results show that the number of KPIs used in projects oscillates between 252 and 552 and captures the complexity of the product as well as the complexity of the project to develop it. Over time the projects change the focus from activities (i.e. what has been done in the project) to product readiness (i.e. what needs to be done in order for the product to be ready for launch). The number, diversity, frequency of updates and the change of focus over time show that the studied company is very mature in the use of KPIs. The results from these investigations supported the company in identifying KPIs which can be dependent on one another (a situation quite possible in such a large data set).

The KPIs for embedded software development are correlated with other KPIs. This implies that, regardless of the applied software development methodology, progress reporting can follow the standard, milestone-oriented progress reporting. It means that companies have some flexibility in changing their ways-of-working within the given frames of strict non-software development projects.

Our further work is an in-depth study to optimize the set of reusable KPIs for an embedded software development project, and to evaluate its feasibility on more cases. This optimal set would help managers to benchmark the projects against each other, quantifying the performance of the project portfolio over time.

Acknowledgements

We would like to thank the Volvo Car Group for letting us conduct the study. We are also grateful to the anonymous reviewers for advice and help in improving the paper.

The research has been partially sponsored by the Swedish Strategic Research Foundation, under grant number SM-13-007.

References

[1] W.S. Humphrey, *Managing technical people: innovation, teamwork, and the software process*. Addison-Wesley Longman Publishing Co., Inc., 1996.

[2] M. Staron, W. Meding, J. Hansson, C. Höglund, K. Niesel, and V. Bergmann, "Dashboards for continuous monitoring of quality for software product under development," *System Qualities and Software Architecture (SQSA)*, 2013.

[3] A.A. De Waal, "The future of the balanced scorecard: an interview with Professor. Dr Robert S. Kaplan," *Measuring Business Excellence*, Vol. 7, No. 1, 2003, pp. 30–35.

[4] "Does measurement theory impact project performance?" *Procedia - Social and Behavioral Sciences*, Vol. 119, 2014, pp. 635 – 644.

[5] A. Neely, B. Marr, G. Roos, S. Pike, and O. Gupta, "Towards the third generation of performance measurement," *Controlling*, Vol. 15, No. 3/4, 2003, pp. 129–135.

[6] International Standard Organization and International Electrotechnical Commission, "ISO/IEC 15939 – software and systems engineering, software measurement process," ISO/IEC, Tech. Rep., 2007.

[7] R.S. Kaplan and D.P. Norton, "Putting the balanced scorecard to work," *Performance measurement, management, and appraisal sourcebook*, Vol. 66, 1995, p. 17511.

[8] M. Staron, W. Meding, J. Hansson, C. Höglund, K. Niesel, and V. Bergmann, "Dashboards for continuous monitoring of quality for software product under development," *System Qualities and Software Architecture (SQSA)*, 2014.

[9] R.S. Kaplan and D.P. Norton, *The balanced scorecard: Translating strategy into action*. Harvard Business Press, 1996.

[10] C. Bentley, *Practical Prince2*. The Stationery Office, 2005.

[11] C. Fornell, "A national customer satisfaction barometer: The Swedish experience," *Journal of Marketing*, 1992, pp. 6–21.

[12] A. Assila, K. Marçal de Oliveira, and H. Ezzedine, "Integration of subjective and objective usability evaluation based on ISO/IEC 15939: A case study for traffic supervision systems," *International Journal of Human–Computer Interaction*, Vol. 32, No. 12, 2016, pp. 931–955.

[13] M. Staron, K. Niesel, and W. Meding, "Selecting the right visualization of indicators and measures–dashboard selection model," in *Software Measurement*. Springer, 2015, pp. 130–143.

[14] M. Staron, W. Meding, and C. Nilsson, "A framework for developing measurement systems and its industrial evaluation," *Information and Software Technology*, Vol. 51, No. 4, 2008, pp. 721–737.

[15] A.A. Mughal, *A Framework for Implementing Software Measurement Programs in Small and Medium Enterprises*, Ph.D. dissertation, University of Otago, 2017.

[16] M. Staron, "Critical role of measures in decision processes: Managerial and technical measures in the context of large software development organizations," *Information and Software Technology*, Vol. 54, No. 8, 2012, pp. 887–899.

[17] P. Runeson, M. Host, A. Rainer, and B. Regnell, *Case study research in software engineering: Guidelines and examples*. John Wiley & Sons, 2012.

[18] B. Azvine, Z. Cui, and D. Nauck, "Towards real-time business intelligence," *BT Technology Journal*, Vol. 23, No. 3, 2005, pp. 214–225.

[19] M. Staron, J. Hansson, R. Feldt, A. Henriksson, W. Meding, S. Nilsson, and C. Hoglund, "Measuring and visualizing code stability–a case study at three companies," in *Software Measurement and the 2013 Eighth International Conference on Software Process and Product Measurement (IWSM-MENSURA), 2013 Joint Conference of the 23rd International Workshop on*. IEEE, 2013, pp. 191–200.

[20] R. Feldt, M. Staron, E. Hult, and T. Liljegren, "Supporting software decision meetings: Heatmaps for visualising test and code measurements," in *Software Engineering and Advanced Applications (SEAA), 2013 39th EUROMICRO Conference on*. IEEE, 2013, pp. 62–69.

[21] M. Staron, W. Meding, C. Hoglund, and J. Hansson, "Identifying implicit architectural dependencies using measures of source code change waves," in *Software Engineering and Advanced Applications (SEAA), 2013 39th EUROMICRO Conference on*. IEEE, 2013, pp. 325–332.

[22] C. Wohlin, P. Runeson, M. Host, M.C. Ohlsson, B. Regnell, and A. Wesslèn, *Experimentation in Software Engineering: An Introduction*. Boston MA: Kluwer Academic Publisher, 2000.

[23] M. Todorović, Z. Mitrović, and D. Bjelica, "Measuring project success in project-oriented organizations," *Management*, Vol. 68, 2013, pp. 41–48.

[24] L. Pilorget, "Process performance indicators and reporting," in *Implementing IT Processes*. Springer, 2015, pp. 177–197.

[25] J. Colin and M. Vanhoucke, "Developing a framework for statistical process control approaches

in project management," *International Journal of Project Management*, 2015.

[26] A. Jaafari, "Project and program diagnostics: A systemic approach," *International Journal of Project Management*, Vol. 25, No. 8, 2007, pp. 781–790.

[27] M.L. Drury-Grogan, "Performance on agile teams: Relating iteration objectives and critical decisions to project management success factors," *Information and Software Technology*, Vol. 56, No. 5, 2014, pp. 506–515.

[28] L. Raymond and F. Bergeron, "Project management information systems: An empirical study of their impact on project managers and project success," *International Journal of Project Management*, Vol. 26, No. 2, 2008, pp. 213–220.

[29] G. Marques, D. Gourc, and M. Lauras, "Multi-criteria performance analysis for decision making in project management," *International Journal of Project Management*, Vol. 29, No. 8, 2011, pp. 1057–1069.

[30] M. Staron and W. Meding, "Transparent measures: cost-efficient measurement processes in SE," in *Software Technology Transfer Workshop, Kista, Sweden*, 2011.

[31] M. Staron, W. Meding, and K. Palm, "Release readiness indicator for mature agile and lean software development projects," in *Agile Processes in Software Engineering and Extreme Programming*. Springer, 2012, pp. 93–107.

[32] H. Sanchez and B. Robert, "Measuring portfolio strategic performance using key performance indicators," *Project Management Journal*, Vol. 41, No. 5, 2010, pp. 64–73.

[33] J. Bilderbeek *et al.*, "R&d performance measurement: more than choosing a set of metrics," *R&D Management*, Vol. 29, No. 1, 1999, pp. 35–46.

[34] F. Spangenberg and D. Göhlich, "Technology roadmapping based on key performance indicators," in *Smart Product Engineering*. Springer, 2013, pp. 377–386.

[35] J.W. Lainhart IV, "COBIT^{TM}: A methodology for managing and controlling information and information technology risks and vulnerabilities," *Journal of Information Systems*, Vol. 14, No. s-1, 2000, pp. 21–25.

[36] V. Basili, J. Heidrich, M. Lindvall, J. Munch, M. Regardie, and A. Trendowicz, "GQM+ Strategies–Aligning Business Strategies with Software Measurement," in *Empirical Software Engineering and Measurement, 2007. ESEM 2007. First International Symposium on*. IEEE, 2007, pp. 488–490.

[37] J. Münch, F. Fagerholm, P. Kettunen, M. Pagels, and J. Partanen, "Experiences and insights from applying GQM+Strategies in a systems product development organisation," in *Software Engineering and Advanced Applications (SEAA), 2013 39th EUROMICRO Conference on*. IEEE, 2013, pp. 70–77.