An Empirical Investigation into Industrial Use of Software Metrics Programs

Prabhat Ram[1], Pilar Rodríguez[2], Markku Oivo[1], Alessandra Bagnato[3], Antonin Abherve[3], Michał Choraś[4] and Rafał Kozik[4]

[1] M3S, Faculty of ITEE, University of Oulu, Oulu 90014, Finland
[2] Faculty of Computer Sciences, Universidad Politécnica de Madrid, 28040 Madrid, Spain
[3] Softeam, 75016 Paris, France
[4] ITTI Sp. z o.o, Poznań and UTP, Bydgoszcz, Poland
{prabhat.ram, markku.oivo}@oulu.fi
pilar.rodriguez@upms.es
{alessandra.bagnato, antonin.abherve}@softeam.fr
chorasm@utp.edu.pl
rafal.kozik@itti.com.pl

Abstract. Practitioners adopt software metrics programs to support their software development from the perspective of either overall quality, performance, or both. Current literature details and justifies the role of a metrics program in a software organization's software development, but empirical evidence to demonstrate its actual use and concomitant benefits remains scarce. In the context of an EU H2020 Project, we conducted a multiple case study to investigate how two software-intensive Agile companies utilized a metrics program in their software development. We invited practitioners from the two case companies to report on the actual use of the metrics program, the underlying rationale, and any benefits they may have witnessed. We also collected and analyzed metrics data from multiple use cases to explain the reported use of the metrics. The analysis revealed improvements like better code review practices and formalization of quality requirements management, either as a direct consequence or as a byproduct of the use of the metrics. The contrasting contexts like company size, project characteristics, and general perspective towards metrics programs could explain why one company viewed the metrics as a trigger for their reported improvements, while the other company saw metrics as the main driver for their improvements. Empirical evidence from our study should help practitioners adopt a more favorable view towards metrics programs, who were otherwise reluctant due to lack of evidence of their utility and benefits in industrial context.

Keywords: Metrics program, process metrics, decision-making.

1 Introduction

With modern software development methods like Agile, the emphasis is on short feedback cycles and quick decision-making, where it is imperative that facts and not mere impressions drive actions [1]. Software practitioners are interested in both min-

ing data for insights and using those insights for decision-making [2, 3], thereby facilitating successful software systems development [4]. Software metrics programs (MPs) can empower practitioners to accomplish the above objectives [5, 6]. Metrics can help generate actionable information to help fulfill an MP's purpose of facilitating decision-making [7, 8]. MPs also enable objective evaluation of software development processes, contributing to continuous improvement and learning in industrial software development context [9]. Current literature reports on successful adoption of MPs [10–13] and characteristics that make metrics actionable [14–16] in Agile software development (ASD). However, empirical investigation into actual MP use for software development in industrial context, and the resulting benefits, remains scarce.

In their literature review, Kupiainen et al. [7] highlighted the lack of empirical evidence demonstrating the use and rationale of MPs in ASD in large industrial context. Some studies provide empirical evidence of success factors for adopting an MP [11–13]. However, the limited study scope prevents discussion of successful MP adoption translating into successful use, especially in ASD. Similarly, studies like [16] and [17] provide evidence of MPs influencing actions in organizations using ASD, but do not detail the motivation of such a use and consequent benefits, if any. Studies like [11–13, 16] and [17] provide a curtailed view of MPs in practice, falling short of discussing MP's use in industrial context and how practitioners benefit from them. We argue that providing such details, supported by clear empirical evidence, can act as a strong motivator for practitioners to adopt a favorable view towards MPs. Moreover, such studies can also provide insights that interested practitioners can follow to replicate successful adoption and use of MPs.

We undertook our research in the context of an EU H2020 Project called Q-Rapids (Project), where the goal was to develop an agile-based, data-driven, and quality-aware rapid software development framework [18]. The Project comprised four software-intensive companies as industrial partners, and we focused our research efforts on the two case companies (CCs) that had progressed to use the Project Solution in their daily work, of which the MP is an integral component. We build upon the findings from our previous studies, where we presented software metrics definition [19] and their successful operationalization [20] at these CCs. The practitioners reported positive outcomes from the use of the MP for software development. These developments motivate the following research question:

RQ: *How do software-intensive companies using Agile software development utilize software metrics in their software development?*

On the back of our collaborations with the CCs, we claim following research contributions. These contributions are enriched by the rationale and validation provided by the CC practitioners invited to contribute to this study.

1. Empirical account of MP use at two contextually different software-intensive companies using ASD.
2. Empirical evidence of MP use for software development and decision-making in software-intensive companies using ASD.

In the remainder of the paper, we discuss background and related work in Section 2, describe the research method in Section 3, and our study's findings in Section 4. We discuss the results in Section 5, threats to our research's validity in Section 6, and conclusion and future research directions in Section 7.

2 Background and Related Work

We first provide a brief background on the Project to aid in comprehension of the study, as we reference different elements and features from the Project throughout the paper. Next, we present the state of the art relevant to our study. Literature documenting use of data to inform an organization's software development process would be relevant for our study. However, since our primary focus is on MPs, we discuss only those literature that center on MPs.

2.1 Q-Rapids Project

The goal of the Project is to develop an agile-based, data-driven, quality-aware rapid software development framework [18]. The objective of this framework (Solution) is to help practitioners make data-driven decisions in rapid cycles. The Solution comprises tools and methods for quality requirements elicitation and management. It also includes a dashboard to monitor indicators concerning product quality, process performance, among others [21].

Our collaboration on the Project centers on definition and operationalization of process metrics. Process metrics measure software development processes. Different process metrics are aggregated into process factors, which are further aggregated into strategic indicators (SIs), collectively constituting the MP. Developers are likely to prefer the lowest level of process metrics, which gives them access to process measurements at ground level. Process factors allow stakeholders like Product Owners (POs) and Project Managers (PMs) to get a refined view of the development process at project and team level. SIs are suitable for upper management, interested in high-level representation of the software development process at their organization.

We focus on the *process performance* SI and the process metrics constituting it. The *process performance* SI measures the performance of an organization's software development process. Among the Solution features, the '*quality alert*' feature [22] carries significance in this study, as we discuss in Section 4.1. Here, once a metric crosses a user-defined threshold, indicating violation of quality goals defined by the use-case Quality Engineers, this feature triggers a quality alert. The alert includes a recommendation, in the form of an abstract quality issue (e.g. quality requirement), to resolve the said violation.

2.2 Related Work

Staron and Meding [13] recommend success factors for MP implementation based on their study of a five-year old MP at Ericsson AB, where they also report that designers and quality managers believe that the MP provides benefits to the management process. Studies by Hall and Fenton [11] and Iversen and Mathiassen [12] focus on success factors for implementing MPs. Similarly, case studies included in the literature review on measurement programs [23] discuss mainly the experience of implementing an MP, and how use of MP facilitated an organization's transition to ASD. In our previous study [20], we focus on presenting the factors for successful operationalization of a metrics program, and a special emphasis is laid on metrics trustworthiness. One common objective missing from these studies is the discussion of actual MP use in an industrial context. Our research addresses this gap by providing empirical evidence demonstrating how practitioners use MPs towards software development, especially in their decision-making.

In their literature review, Kupiainen et al. [7] call for more empirical studies to explore the rationale and use of MPs, especially in large industrial context. Although most existing studies present initial emerging results of using an MP, lacking empirical evaluation in industrial context, there are few exceptions. A study by Dubinsky et al. [24] report on the use of an MP at an extreme programming (XP) development team of the Israeli Air Force. The authors conclude that use of metrics could lead to more accurate and professional decision-making. Díaz-Ley et al. [25] studied a measurement program (synonymous with metrics programs) targeted towards small-and-medium-size enterprises (SMEs), and found that use of metrics can help practitioners define measurement goals that are well aligned with their organization's maturity. Port and Taber [16] provide empirical evidence of MP use and their actionability in a large industrial context. With the help of metrics and analytics programs, the authors illustrate the supporting role an MP played in strategic maintenance of a critical system at NASA's Jet Propulsion Laboratory. Similarly, Vacanti and Vallet [17] conducted a case study at Siemens Health Services (SHS), and presented results of an MP's actionability, helping SHS increase productivity and improve process performance. These studies highlight MP use and its potential for influencing actions in software development in industrial context, but empirical evidence for the same is limited to only one case company. Furthermore, the rationale that underlie the reported use of the metrics and their alleged benefits lack explicit validation by the practitioners involved. Our research targets the common objective of providing empirical evidence of MP use in large industrial context. However, our research also includes multiple CCs and use cases (UCs) to argue the said objective. Furthermore, we support our claims by inviting the involved practitioners to validate them and to provide additional insights, especially the rationale that drive their MP utilization.

3 Research Method

Following the guidelines recommended by Runeson and Höst [26], we conducted a multiple case study to answer the RQ. In addition, we invited practitioners from each CC to validate our findings and provide supporting rationales.

3.1 Research Context

In order to understand better the MP use and the underlying rationale, we describe both the software development context and the Solution context at the two CCs.

Software development context. The following table characterizes the two CCs' software development context:

Table 1. Case Company Characteristics

Parameters	Case Company 1	Case Company 2
ID	CC1	CC2
Size	Large	SME
Domain	Commercial services and solutions	Multi-industry
Development method	Customized Agile	*ScrumBan* & *ScrumBut*
Use case(s)	Software modeling tool	Warehouse Mgmt. System
Length of Solution use	~ 2.5 years	~ 2 years
Use case team size	9	15

Case Company 1 (CC1) is a large-size company (>900 employees), with the goal of using the MP to improve the quality of its ASD process through early detection of anomalies in their development. CC1 uses various software development methods that adhere to Agile principles. Among other solutions, CC1 develops modeling tool for model-driven development. Part of a 25-year-old product line, the company has multiple releases of this tool in the market. CC1 has used the Solution in the course of development of the past four releases of the tool. In our study, we utilize data from three product releases (UC1.1 – UC1.3) that were developed during the course of the Project.

Case Company 2 (CC2) is an SME type consulting company (around 100 employees), developing solutions for multiple industrial and application domains (e.g. administration, utilities, e-Health, etc.). The company has its own process for acquiring functional and quality requirements, and the initial mockups and user stories collected during this process forms the basis for their development process. Due to these exceptions, the company reports its development method as *ScrumBut*. Similarly, *ScrumBan* refers to the company's iterative software development, and its use of Kanban board to monitor backlogs. Currently, the company is in the process of going agile on a large-scale, and uses the abovementioned customized Agile approach for

project management. The company aims to use the MP to allow its developers to anticipate design issues, security issues, and platform limitations. After piloting the Solution on mostly finished projects, the company used it in a project to develop an enterprise class integrated software system for managing warehouses. We focus on this project (UC2.1) for our study.

Solution Context. On the back of their experience of using generic metrics (e.g. SonarQube[1] code quality metrics), CC1 were convinced that an MP could be useful only if it was adapted to their specific context, with respect to their processes and the data available. The MP allowed CC1 practitioners to capitalize and analyze the historical data, and identify problems and obstacles to their processes. This is evident in their choice of process metrics, available in *Appendix*[2]. CC1 found that involvement of MP target users played a significant role in their successful operationalization of the MP. Their involvement and feedback helped CC1 get the results that they deemed reliable. This was accomplished only after adapting the MP to their context, leading to growth in target users' confidence in the MP. As a result, the CC1 UC development team utilizes the metrics in their regular meetings to discuss new releases, and plan steps for the next product releases.

Informed by their positive and formative experience from the pilot UC, CC2 continued to use the Solution in their next UC, albeit with customizations that are exclusive to this company. Influenced by their use of SonarQube code quality metrics, CC2 had reservations about the MP's potential to induce process improvements. They viewed it as a tool that can help monitor the process, and can be of value if and only if it were tailored to CC2's projects. This perspective largely dictated their choice of metrics. For CC2, MP operationalization was a success because of three major factors, (i) some process metrics provided additional value (unavailable elsewhere) in understanding the development process, (ii) the MP was first tested and introduced gradually to complement existing processes, and (iii) the MP facilitated and enhanced CC2's culture of transparency. With respect to the second factor, CC2 evaluated the MP's usefulness by conducting a retrospective session. The objectives were to explore the relevant process metrics used in the first six months of UC2.1, get feedback on the reasons of the impacts of the process metrics, and document the results of these impacts in a template. Consequently, CC2 arrived at a list of process metrics considered most valuable by its practitioners. These metrics were related to estimation efforts, bug density, issues velocity, and bug correction performance. In terms of transparency, the MP helped anchor PO's opinions and decisions in actual data, which the developers could verify and validate by using the Solution dashboard.

[1] https://docs.sonarqube.org/latest/user-guide/metric-definitions/

[2] https://zenodo.org/record/3953067#.X2DSo3kzZaQ

3.2 Data Collection

The CCs had been sharing process metrics data with the Project researchers on a monthly basis, along with a short report on their use of the Solution. In the course of several follow-up interactions with the CCs, we also learned that they had been utilizing MP to undertake process improvements. The following table provides context for the data we collected for this study:

Table 2. Data collection context

Parameters	CC1	CC2
Data period	Oct. 2018 – Sep. 2019	Nov. 2018 – Aug. 2019
Use Case	UC1.1 – UC1.3	UC2.1
Type of data	Process metrics	
Focus	*Process performance* (SI)	

Our decision to focus on process metrics data was based on the objective of evaluating the use of MP, as reported by the CCs, and because we were responsible for implementing only process metrics. Both CCs reported an overall improvement in the *process performance* SI, which is automatically computed and collected on a daily basis. For example, the *process performance* SI in CC1 is the average of the three process factors of *tasks' velocity, testing performance, and testing performance.* These individual process factors are, in turn, the average of the process metrics that constitute them. The same logic applies for SI computation for CC2. These data provide the evidence necessary to support the reported MP use by the two CCs. In addition, the data also provide a quantitative underpinning to the rationale provided by the CC practitioners invited to contribute to this study.

3.3 Data Analysis

The following table presents the analysis approach we adopted for this study. The focus is on analyzing the *process performance* SI data in order to explain CCs' metrics utilization in their software development, especially towards decisions for process improvements. SI data are the only quantitative evidence available to draw any legitimate conclusion towards the reported use of the Solution, and to anchor the subsequent discussion by the invited practitioners.

Table 3. Data Analysis approach

Parameters	CC1	CC2
Analysis	*Kruskal-Wallis Test*, and *Pairwise Mann-Whitney U Test*	Trend analysis

In CC1, the extent of MP use across the three UCs evolved, as the practitioners customized and refined the process metrics to reflect their way of working. The SI data does not follow normal distribution, so we use *Kruskal-Wallis Test*, also known as one-way ANOVA by rank. It is a non-parametric test that can help us assess whether the difference in the *process performance* SI measured across the three UCs is statistically significant. In the event that it is, we use *Pairwise Mann-Whitney U-Test* as a post-hoc analysis to determine the specific UC that is statistically significant from others. We did not use *Kruskal-Wallis Test* in case of CC2, because the data comes from just one UC. Instead, we perform trend analysis on the UC2.1 SI data. Post analysis, we invited practitioners from each CC to review our findings, and provide rationale to support our claims and their reported MP use.

4 Findings

We first present the empirical evidence of the MP use for software development at each CC, based on the reported use of metrics for specific interventions and improvements. Here, the invited practitioners provide the necessary background, the actual use of the metrics for the above-stated purpose, and the rationale for such use, especially with respect to their decision-making. Next, we analyze the process metrics data to strengthen the above claims, providing a quantitative background to the reported use and the reported benefits, if any.

4.1 CC1

The MP helped CC1 identify the blocking points that could cause potential delays in their release. For example, CC1 used the '*non-blocking files*' metric to identify problems blocking their development tasks, critical for development features for their upcoming release, and prioritized the said development activities. In addition to bottleneck identification, the MP also facilitated process improvements, apparent from the results of using the '*critical issues ratio*' process metric. Here, however, the metric's influence was supplemented by the '*quality alert*' feature. The above metric triggered an alert, which recommended a *quality requirement*. The PM accepted the said quality requirement, and proposed it as a development task to address the problem both the metric and the alert indicated. This formalized quality requirements management is an improvement over CC1's past ad-hoc resource mobilization to address quality-related issues. Based on the above two instances of MP use, CC1 managed to improve their development process by improving their product quality, and optimized their effort to manage that quality.

Overall, and reflecting the improvements described above, CC1 reported an improvement of 10% - 20% in their *process performance* SI since they started using the Solution, which includes the MP. UC1.1 data available is from the period when its release was due by around two months. For consistency, we used UC1.2 and UC1.3

data from similar periods. We also excluded data from the process factor '*tasks velocity*' due to reliability issues. The process metrics, and their interrelationships with process factors and the SI specific to CC1, is available in *Appendix*. The following table gives a snapshot of the data we used to support the reported MP use and improvements:

Table 4. CC1 data context

Use Case	Data	Period	Total Data points	Solution Use
UC1.1	*Process performance* SI	Two months before the release date	186	Low
UC1.2				Medium
UC1.3				Medium

The extent of *Solution Use* was determined by the Solution's maturity and availability of its different features. This also includes the perceived reliability of the MP data. CC1 Champion confirmed the above labelling of the extent of *Solution Use*. The following table provides descriptive statistics of *the process performance* SI from the three UC datasets, followed by their boxplot visualization in **Fig. 1**. Each UC dataset comprises the SI data computed and collected on a daily basis, and is from a period of two months prior to the release date.

Table 5. Descriptive statistics of CC1 dataset

Solution Use	*N*	Mean	Standard deviation	Median
UC1.1	61	0.829	0.0429	0.845
UC1.2	63	0.854	0.0258	0.855
UC1.3	62	0.907	0.0291	0.920

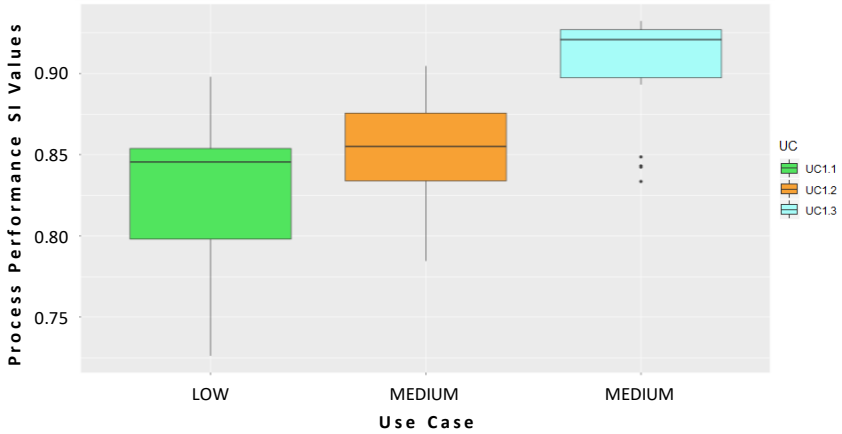

Fig. 1. Box-plot for the CC1 *process performance* SI

In the above box-plot, the Y-axis represents the SI values. The X-axis corresponds to the extent of *Solution Use* for the three respective UCs. The chart demonstrates that the median SI across the three UCs increased as the use of the Solution increased at CC1. Despite the extent of *Solution Use* for both UC1.2 and UC1.3 being the same (*Medium*), the chart suggests performance difference between the two UC samples. We analyzed the SI data using *Kruskal-Wallis Test* with the null hypothesis (H_0): the samples come from the same population, and alternative hypothesis (H_a): the samples do not come from the same population. The results from the test are as follows:

Table 6. Kruskal-Wallis Test

Kruskal-Wallis Test for CC1 *Process performance* SI	
K (*observed chi-square value*)	85.27
K (*critical chi-square value*)	5.99
df	2
p value	<0.0001
alpha	0.05
epsilon squared (*effect size measure*)	0.46

With the *critical chi-square value* (5.99) less than the *observed chi-square value* (85.27), and with the computed *p* value less than the significance level alpha = 0.05, we reject H_0 and accept H_a. This suggests that the SI values in three UCs have statistically significant performance difference from each other for at least one UC sample, corresponding to a strong effect size (epsilon squared, e^2 =.0.46). In order to determine which UC sample is significantly different, we performed pairwise comparison using *Pairwise Mann-Whitney U Test*. The result of this post-hoc analysis test is as follows:

Table 7. Pairwise Mann-Whitney U Test results

Pairwise Mann-Whitney U Test	**UC1.1**	**UC1.2**
UC1.2	0.011	
UC1.3	<0.0001	<0.0001

The above table lists *p* values as the result from the comparison among UCs. All the computed *p* values for these comparisons are less than the significance level alpha = 0.05. This suggests that there is a statistically significant difference among UCs. More specifically, UC1.2 is significantly different from UC1.1, and UC1.3 is significantly different from both UC1.1 and UC1.2. This significant difference coincides with the increased use of the Solution across these UCs.

In view of CC1's MP use, informed by their approach to the MP and experience with it so far, CC1 practitioners maintain that the MP was one of the contributors to influence their software development, resulting in benefits like identification of blocking points and formalization of their quality requirements management process. Matu-

ration effect could be one of the reasons why despite the medium *Solution Use* in both UC1.2 and UC1.3, the difference in the *process performance* SI for the latter was statistically significant from the other two UCs. CC1 finds that the MP gives the practitioners a *'behind the scene'* of their development process, broadening their overall perspective. As a result, CC1 views MP more as a decision-support tool, rather than a control tool, which aligns well with the perspective they harbored for MPs in general from the start of the Project.

4.2 CC2

The MP helped improve CC2's code review process, by allowing medium-experienced developers address *merge requests*, a task earlier reserved only for experienced developers. The MP also allowed the PO and Senior Managers (SMs) to identify that four days is the optimal reported effort spent on a task, which in turn improved their developers' efficiency. CC2 was interested in POs and SMs making quick team-oriented decisions without involving too many stakeholders like the CEO or the company board. The MP made this possible, as now the CC2 practitioners had the means to verify and validate PO and SM's actions. The trust CC2 managed to build for the MP is evident in its use in the weekly Scrum meetings, to learn to improve their way of working, motivate the team, and identify problems and find solutions for them.

Similar to the results from the pilot UC, CC2 reported an overall improvement even for their UC2.1 *process performance* SI. The data corresponds to the MP use throughout UC2.1, but we excluded the data for the '*resolved issues throughput*' process metrics due to reliability concerns. The following table provides an overview of the data we used to conduct the trend analysis, illustrated in **Fig. 2**:

Table 8. CC2 data context

Use Case	Data	Period	Total Data points
UC2.1	*Process performance* SI	Throughout UC2.1	225

CC2 Champion demarcated the three *Periods* in the chart to highlight the exceptional *Period 2*. Certain eventualities at the company impacted CC2's software development performance for a short period, which is reflected in the downward trend of *Period 2*. Otherwise, the chart indicates an upward trend, suggesting improvement in UC2.1's *process performance* SI as the Solution use increased, relative to the use in the pilot UC.

Process metrics from the MP made possible the important changes in CC2, like enabling a suitable team management culture. In addition to other in-house metrics CC2 developed, the process metrics from the MP facilitated PO's understanding of the process dynamics. Based on the MP, CC2's decisions like increasing the number of

developers to perform merge requests and improving effort estimation, reveals the practitioners were relying completely and only on the MP for the above improvement decisions. This claim is further supported by the improvement seen in UC2.1 SI, as depicted in the upward trends in Period 1 and Period 3 in **Fig. 2**.

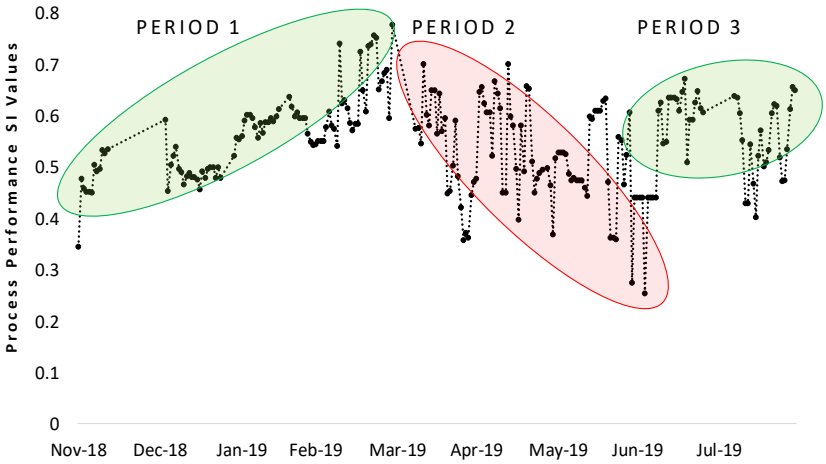

Fig. 2. UC2.1 *process performance* SI trend

5 Discussion

Based on the two CCs' reported MP use, their underlying rationales, and the alleged role of the metrics in their decision-making, we argue that the CC practitioners used the MP as either a trigger or the main driver for their software development. We discuss the results from these two perspectives and answer the RQ.

5.1 Metrics program as a trigger

Based on the reported MP use by CC1, informed by their experience and their general approach to MPs, the practitioners viewed metrics as a trigger that directed them towards taking certain actions. CC1 practitioners believe a standalone MP is not enough to influence actions and drive development. Other contributing factors play an equally important role. They subscribe to the idea that MPs, and the underlying metrics, are not '*magic*' that can lay out clear and concise action to be taken [14]. The MP can only guide them, provided its data are corroborated using different means like the original data source or an authority at the organization. The need for corroboration is an implicit requirement for evaluating MP data reliability [2]. Any meaningful improvements and development driven by MPs can only be a result of several other factors working in harmony [27]. This principle is visible in CC1's use of '*critical issues ratio*' metric and the '*quality alert*' feature, as part of their new and formalized quality requirement management process.

CC1 is a large-size company, which may explain their reluctance at relying solely on the MP for making development decisions. With respect to project characteristics, it is difficult to draw any conclusion, as every single CC1 UCs involved development of different versions of the same modeling tool. However, based on [7], we argue that at least company size, if not project characteristics, moderated the MP's potential to influence software development at CC1. Furthermore, CC1's perspective towards the MP remained, largely, unchanged throughout the Project, which further dictated their utilization of the MP as only a tool to inform their decision-making. Their original perspective of the MP being only a decision-support tool is also in line with the claim of CC1 using the MP only as a trigger.

5.2 Metrics program as a main driver

The retrospective sessions helped CC2 establish an organization-wide trust in the MP. The resulting transparency compelled the PO and SM to use the MP for improving '*merge requests*' and effort estimation processes, which led to positive outcomes. Therefore, we argue that CC2 are proponents of using MP as a main driver in their software development. This stance is predicated on conditions like adapting the MP to their context, target user feedback, and transparency.

On the back of better visibility and overall transparency, CC2 now completely relies on the MP to carry out improvements. This is compatible with the findings in [17], where specific interventions inferred from the *flow*[3] metrics led to an increase in productivity and process performance at SHS. Furthermore, CC2 also achieved the goal of equipping its POs and SMs with the responsibility to take decisions independent of other stakeholders. The retrospective sessions were instrumental in convincing the CC2 of the MP's resourcefulness for achieving the said goal. With the MP now adapted to CC2's context, its practitioners are confident in using the MP as their main driver towards everyday software development, and even undertaking decisions for improvements. This clearly highlights a diametric shift in CC2's earlier view of MP being useful mainly as a monitoring tool.

6 Threats to validity

We report on the threats to our study's validity based on the guidelines recommended by Runeson and Höst [26].

The study is designed based on the MP and its constituents of process metrics, process factors, and SIs, which have been verified in theory and in practice in our previous studies [19, 20]. However, there is potential for misrepresentation of the results of the quantitative analysis, which threatens the study's construct validity. To mitigate this threat, we discussed and verified these results with the CC practitioners, particu-

[3] Workflow metrics such as Work in Progress, Cycle Time, and Throughput

larly the Project/UC champion. Furthermore, we invited them to contribute to this study, and validate the results and the claims derived from them.

Other confounding factors could have influenced our results, affecting the internal validity of our claim regarding how each CC used the MP. For example, improvement in UC1.3 SI values could be due to maturation effect. We have tried to mitigate this threat by allowing the practitioners from the corresponding CC to validate these claims by providing the underlying rationale. Furthermore, we excluded the data that could interfere with the legitimacy of the results, and kept the CC practitioners in the loop of every decision concerning data collection and processing.

The study involves only two software-intensive companies, each with their distinct context, which affects the external validity of our study. However, we have provided a detailed context for each CC, and used that to shape our discussion on their specific MP use for software development and improvement decisions. Therefore, our findings may be applicable to organizations that are similar in context to any of the two CCs. Moreover, rather than a rulebook, our study can serve as a starting point for interested organizations. Additionally, our overarching goal is to trigger further investigations on the research topic and gaps we have attempted to address here.

Multiple researchers and practitioners have helped elaborate and validate the findings from the study. However, only one researcher was involved in metrics data collection, which may affect the reliability of our study.

7 Conclusion

The state of the art provides limited empirical evidence for metrics programs use in ASD in industrial contexts, especially for decision-making. However, studies that provide empirical evidence of metrics programs use in large industrial settings, supported by their usage rationale and consequent benefits, remains scarce. In the context of the EU H2020 Q-Rapids Project, we have tried to address this research gap.

We collaborated with two software-intensive companies, and reported on their metrics programs use for software development, including decisions made towards process improvements. Analyzing process metrics data from the two case companies, we reinforced the empirical evidence to support the rationale provided by the practitioners for their reported use of metrics programs for software development and decision-making. Company size and perspective towards MP's potential for software development are the two probable distinguishing criteria explaining their use at the two case companies. For a large-size company with a cautionary perspective, a metrics program can only act as a trigger for software development and decision-making. In contrast, for SMEs, it can act as the main driver, provided company-specific conditions like adapting to their context, target user feedback and transparency are met.

Future work could include evaluation of several use cases across multiple large-size companies and SMEs to improve generalizability of our findings.

Acknowledgments

This work is a result of the Q-Rapids Project, funded by the European Union's Horizon 2020 research and innovation program, under grant agreement No. 732253.

References

1. Liechti, O., Pasquier, J., Reis, R.: Beyond dashboards: On the many facets of metrics and feedback in agile organizations. In: Proceedings - 2017 IEEE/ACM 10th International Workshop on Cooperative and Human Aspects of Software Engineering, CHASE 2017. pp. 16–22 (2017).
2. Staron, M., Meding, W.: Ensuring reliability of information provided by measurement systems. In: Software Process and Product Measurement, International Conferences IWSM 2009 and Mensura 2009. pp. 1–16. Springer, Berlin, Heidelberg (2009).
3. Yang, Y., Falessi, D., Menzies, T., Hihn, J.: Actionable analytics for you. IEEE Softw. 35, 51–53 (2018).
4. Bird, C., Murphy, B., Nagappan, N., Zimmermann, T.: Empirical software engineering at Microsoft Research. Proc. ACM Conf. Comput. Support. Coop. Work. CSCW. 143–150 (2011).
5. Menzies, T., Zimmermann, T.: Software analytics: So what? IEEE Softw. 30, 31–37 (2013).
6. Zhang, D., Han, S., Dang, Y., Lou, J.G., Zhang, H., Xie, T.: Software Analytics in Practice. IEEE Softw. 30, 30–37 (2013).
7. Kupiainen, E., Mäntylä, M. V., Itkonen, J.: Using metrics in Agile and Lean software development - A systematic literature review of industrial studies. Inf. Softw. Technol. 62, 143–163 (2015).
8. Staron, M., Meding, W.: Transparent measures : cost-efficient measurement processes in SE. In: Software Technology Transfer Workshop. pp. 1–4. , Kista, Sweden. (2015).
9. van Solingen, R., Berghout, E.: Integrating goal-oriented measurement in industrial software engineering: industrial experiences with and additions to the Goal/Question/Metric method (GQM). Proc. Seventh Int. Softw. Metrics Symp. 246–258 (2001).
10. Mendonça, M.G., Basili, V.R.: Validation of an approach for improving existing measurement frameworks. IEEE Trans. Softw. Eng. 26, 484–499 (2000).
11. Hall, T., Fenton, N.: Implementing effective software metrics programs. IEEE Softw. 14, 55–64 (1997).
12. Iversen, J., Mathiassen, L.: Cultivation and engineering of a software metrics program. Inf. Syst. J. 13, 3–19 (2003).
13. Staron, M., Meding, W.: Factors determining long-term success of a measurement program: An industrial case study. e-Informatica Softw. Eng. J. 1, 7–23 (2012).

14. Croll, A., Yoskovitz, B.: Lean Analytics: Use Data to Build a Better Startup Faster. (2013).
15. Buse, R.P.L., Zimmermann, T.: Information Needs for Software Development Analytics - Microsoft Research. MSR Tech Rep. 2011-8. 1–16 (2011).
16. Port, D., Taber, B.: Actionable Analytics for Strategic Maintenance of Critical Software: An Industry Experience Report. IEEE Softw. 35, 58–63 (2017).
17. Vacanti, D., Vallet, B.: Actionable Metrics at Siemens Health Services. (2014).
18. Franch, X., Ayala, C., López, L., Martínez-Fernández, S., Rodríguez, P., Gómez, C., Jedlitschka, A., Oivo, M., Partanen, J., Räty, T., Rytivaara, V.: Data-driven requirements engineering in agile projects: the Q-rapids approach. Proc. - 2017 IEEE 25th Int. Requir. Eng. Conf. Work. REW 2017. 411–414 (2017).
19. Ram, P., Rodriguez, P., Oivo, M.: Software process measurement and related challenges in agile software development: A multiple case study. In: International Conference on Product-Focused Software Process Improvement. pp. 272–287. Springer Verlag (2018).
20. Ram, P., Rodriguez, P., Oivo, M.: Success Factors for Effective Process Metrics Operationalization in Agile Software Development : A Multiple Case Study. In: Proceedings of the 2019 International Conference on Software and System Process (2019).
21. López, L., Martínez-Fernández, S., Gómez, C., Choraś, M., Kozik, R., Guzmán, L., Vollmer, A.M., Franch, X., Jedlitschka, A.: Q-rapids tool prototype: Supporting decision-makers in managing quality in rapid software development. In: Lecture Notes in Business Information Processing. pp. 200–208. Springer, Cham (2018).
22. Oriol, M., Seppänen, P., Behutiye, W., Farré, C., Kozik, R., Martínez-Fernández, S., Rodríguez, P., Franch, X., Aaramaa, S., Abhervé, A., Choras, M., Partanen, J.: Data-Driven Elicitation of Quality Requirements in Agile Companies. In: Communications in Computer and Information Science. pp. 49–63. Springer Verlag (2019).
23. Tahir, T., Rasool, G., Gencel, C.: A systematic literature review on software measurement programs. Inf. Softw. Technol. 73, 101–121 (2016).
24. Dubinsky, Y., Talby, D., Hazzan, O., Keren, A.: Agile metrics at the Israeli Air Force. In: Agile Development Conference (ADC'05). pp. 12–19. IEEE Comput. Soc (2005).
25. Díaz-Ley, M., García, F., Piattini, M.: Implementing software measurement programs in non mature small settings. In: Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). pp. 154–167 (2008).
26. Runeson, P., Höst, M.: Guidelines for conducting and reporting case study research in software engineering. Empir. Softw. Eng. 14, 131–164 (2009).
27. Meneely, A.: Actionable metrics are better metrics. In: Perspectives on Data Science for Software Engineering. pp. 283–287. Elsevier (2016).