Information and Software Technology xxx (xxxx) xxx–xxx

Contents lists available at ScienceDirect

Information and Software Technology

journal homepage: www.elsevier.com/locate/infsof

Improving the experience for software-measurement system end-users: A story of two companies

Haoyu Wang[a], Regina Hebig[*,b,c]

[a] Ericsson, Shanghai, China
[b] Chalmers, Gothenburg, Sweden
[c] University of Gothenburg, Gothenburg, Sweden

ARTICLE INFO

Keywords:
Measurement systems
End-user experience
Performance

ABSTRACT

Context: Software measurement systems are used in large companies to provide developers with up-to-date feedback and metrics.
Objective: However, the front-ends of these systems are often not ready to provide a real-time experience for the end-users, who sometimes have to wait minutes before visualizations are provided.
Method: In this paper, we present the case studies of two large international companies and compare alternative technological setups for measurement system front-ends used and explored within these companies. We use a publicly available data-set for a performance evaluation and to analyze the results.
Results: For both companies we found significant performance differences between the alternative setups. However, in one of the companies these differences are not there for large data-sets. Furthermore, we found that not all setups enable the visualization of the newest available data.
Conclusions: Our results indicate that the choice of the visualization component has a larger impact on the performance than the choice of the data storage. However, companies are also willing to invest into setups that ensure that visualized measures are always up-to-date.

1. Introduction

Software development is a hugely complex task that requires developers to consider and balance aspects such as quality of code and technical depth, while they develop new functionality and coordinate their teamwork. To support the developers, companies strive to provide them with the right information in the right way. For example, Ericsson aims to provide developers and managers with constant information about the current development and trends in the number of defects in their systems as they consider the quality of the code as a primary concern. Similar needs exist for decision making in other sectors that rely on measurement visualization, as for example in the financial sector.

Companies that operate on a large scale can set up software measurement systems to automate the regular collection of hundreds of metrics [1]. These systems are bothered with providing measurement as a service [2] to their users, thus, enabling them to focus on their main tasks. Research in this direction is concerned with the question about the quality and reliability of the provided data [3].

Such measurement systems consist of a front-end and a back-end. The latter is responsible for collecting and extracting measurements as well as loading them into the storage to ensure regular updates of the data. The front-end, however, is responsible for analyzing/aggregating the data to provide meaningful visualizations to the end-user. Thus, the real-time experience of the end-user depends on the question how fast the front-end can access and analyze the updated data points to refresh the visualization.

Problem statement. However, the large scale and regularly updated measurements imply the challenge to also analyze the data in a real-time manner and to display only up-to-date information, as already emphasized by Basili [4]. The idea of real-time measurements can be found in many other disciplines. For example, in medicine real-time visualization and analysis of measurements is highly relevant [5]. Also in the area of business intelligence the real-time analysis and visualization of data and measurement is one of the core topics addressed in the last years [6,7]. This real-time experience for the

[*] Corresponding author.
E-mail addresses: haoyu.h.wang@ericsson.com (H. Wang), hebig@chalmers.se, regina.hebig@cse.gu.se (R. Hebig).

https://doi.org/10.1016/j.infsof.2018.06.006
Received 25 January 2018; Received in revised form 12 June 2018; Accepted 16 June 2018

user is also more and more focus of companies using measurement systems. However, the large amount of measures can slow down the creation of visualizations dramatically. Also, even though measures might be up-to-date in the storage, not every technological setup allows to always have the newest data visualized.

In this paper we investigate cases of two companies who experimented with and changed the technologies used the front-ends of their measurements systems, with the clear goal to improve the users' experience. We compare the setups and analyze pain points that let to change decisions, by comparing properties of the set-ups including their performance.

Research questions. To better support the users, there was a strong desire in both companies to provide a real-time experience in the front-end. Therefore, new technologies for implementing the front-end were tried out in both cases. In this paper, we report on these new setups and the reached differences with regards to the speed in which visualizations can be rendered in the front-end. In doing so, we aim to answer the following research questions:

- RQ1 What changes in technology setups lead to the performance improvement in a measurement system's front-end?
- RQ2 Is the improvement sufficient to impact the end-user experience?

Extension. This paper is an extension of the conference paper [8], which already included the Ericsson case study. For this paper we added the case of the Financial Software Company and extended the discussion to compare decision making of the two cases.

Tour. In this paper, we present the cases of two companies that changed the technology setups of the front-ends of their measurement systems. Section 2 summarizes related work. In Section 3 we present the method that we applied for each case to compare the performance of the applied setups. The setups of the companies and the assessed measurements are presented in Sections 4 and 5). We discuss the results of the comparison with regards to our research questions in Section 6. Finally, we conclude in Section 7.

2. Related work

In the following we discus related work on awareness support and automated measurements and measurement programs as well as works on the performance of data visualizations, in general and with a special focus on Tableau.

Awareness support. Many works investigate the need for sharing awareness amongst teams. For example, Treude et al. [9] performed an empirical study with large development teams and found out that dashboards have a crucial role during definition of task priorities and help to identify bottlenecks. Their interviewees pointed out the need to constantly report information based on large data collections. Treude et al. also found that dashboards change overtime with the progress of the project.

Wallace et al. [10] used an experimental setup by giving teams tasks to investigate the role of shared displays. They found out that setups with status displays can be used best to monitor a teams progress, while large shared displays can also be used by teams to support conversations. Furthermore, the researchers found that teams quickly use the display to synchronize, using gazes and body language. The importance of dashboards and visualizations was further confirmed in an interview study with Bugzilla, performed by Baysal et al. [11]. They found that developers value the updates on change/progress that can be seen when regularly checking the screen.

From a systematic literature review performed by Kupiainen et al. [12] we know that agile teams use a multitude of metrics and visualizations during development, including for example, build status, technical debt in effort, number of automated passing test steps, or number of defects/velocity.

Automated measurements and measurement programs. Thus, measurement visualizations as part of dashboards and shared screens are common in industry. One example of an early work that provides a dashboard for awareness support is FastDash [13]. The Dashboard provides interactive visualization with the goal to allow teams identifying potential conflict situations.

Since then many works consider a larger, company-wide scale of awareness sharing. This contains on the one hand works that are concerned with automating single measures, such as μ_c*ROSE* by Diab et al. [14] or the work by Gonultas et al. [15], which both aim at automating the measure COSMIC function points.

On the other hand, there are works that are concerned with the instantiation of company wide measurement systems. For example Diaz et al. [16] study how the MIS-PyME framework for defining measurement programs can be applied to medium sized companies. They found that - comparable to the observations of Treude et al. [9] - a challenge is to adapt and adjust the frameworks and measurements over time as the company matures.

Another example of a measurement framework was introduced by Staron et al. [1]. They performed a study on an Ericsson measurement program to identify success factors for these frameworks. Amongst others, they found that reliability of the system is crucial in terms of two aspects: 1. that the provided information is correct and 2. that the provided information is up-to-date [17]. They also provide a first step towards ensuring reliability of the provided information by applying a notion of quality on the measures [18].

Performance of data visualizations. Godfrey et al. [19] performed a survey on research about interactive visualization of large datasets (focusing on querying systems). They observe the need for scalable visualizations, which is addressed by research that uses sampling or incremental visualization to deal with the data size. Two examples for that are the works of Liu et al. [20] and Rahman et al. [21]. Liu et al. [20] use a filtering and sampling technique combined with parallel processing to make visualizations scalable. They report on having managed to keep the performance stable even if data size rises to thousands or even billions of records. Rahman et al. [21] tackle the problem, by working with incremental and interruptible visualizations that show a first view on the data based on a sample and constantly increase the degree of detail by adding more information, until the user stops the process.

Tableau optimizations. Furthermore, there are a lot of studies on commercial systems such as Tableau. A study concerning the load of Tableau Public and the system Many Eyes was performed by Morton et al. [22] in form of a longitudinal study. They measured that 84% of the visualizations created in Tableau Public took less than 2 s to load. However, there is nonetheless work going on, concerning the large scale data. Wesley et al. [23] presented in 2011 the new tableau data engine and did a performance evaluation to compare the data engine with the Firebird database, MonetDB and TDE. They show that the selection of the database can lead to 1–3 orders of magnitude performance increase. For MonetDB and TDE they measured execution times mostly below 2 s. In a later work Welay et al. [24] describe compression techniques and optimizations in tableau, such as the use of column stores - focusing mostly on the querying aspects. Finally, Terlecki et al. [25] uses catching and prefetching for improving performance within Tableau. However, they do not provide a performance comparison.