Gothenburg University Publications

Measurement-as-a-Service - A New Way of Organizing Measurement Programs in Large Software Development Companies

This is an author produced version of a paper published in:

International Conference on Software Measurement (Mensura)

(article starts on next page)

Measurement-as-a-Service – A New Way of Organizing Measurement Programs in Large Software Development Companies

Miroslaw Staron[1] and Wilhelm Meding[2]

[1] Computer Science and Engineering, University of Gothenburg, Sweden
`miroslaw.staron@gu.se`,
[2] Ericsson AB, Sweden
wilhelm.meding@ericsson.com

Abstract. Modern software development companies focus on their primary business objectives, delivering customer value and customer satisfaction which often leads to prioritization of core business areas over such areas as measurement. Although the companies recognize the need and importance of software measurement, they often do not have the competence and/or time to focus on software measurement. In this paper we address the challenge of optimizing the measurement processes in modern companies by using cloud computing and by providing measurement (process) as a service for core business of the companies. Similar to the concept of Software-as-a-Service we define the concept Measurement-as-a-Service and describe how to organize a measurement program according to this definition. The Measurement-as-a-Service concept is well-aligned with measurement programs developed according to ISO/IEC 15939 and can help the companies to increase the benefits obtained from the efficient use of metrics.

Key words: cloud, measurement systems, measurement program

1 Introduction

Modern software development companies focus on their core businesses and on delivering customer value and aligning their processes towards that. Measurement programs in such companies are often designed to support these goals, but they do not form the core business areas and as such can be optimized in a different manner [JA97] [SMKN10]. Instead of focusing directly on the customers, the measurement programs are focused on internal stakeholders – which represent either the external customers or internal roles in the company (e.g. quality management) [SM09b].

This kind of evolution of the software business model creates new opportunities for the evolution of the measurement programs at the software development companies – centralizing the development and delivery of the measurement programs in companies. As prescribed by the ISO/IEC 15939 (Systems and Software

Engineering – Measurement Processes) standard [OC07, SMN08] the measurement programs should include the set of measurement systems and the infrastructure supporting building and disseminating the knowledge base of software measurement.

In this paper we address the following research problem – *How to support the company's core business processes by optimizing the sharing of measurement competence across different units?* To address this research question we introduce the term *Measurement-as-a-Service, MaaS*. MaaS is a measurement licensing and delivery model in which metrics are licensed on a subscription basis, centrally hosted, collected and delivered on demand. This concept is similar to the concept of Software-as-a-Service and Platform-as-a-Service. As the definition of Software-as-a-Service describes the licensing, delivery model and value proposition for centrally hosted software available on the web, we propose to use measurement in the same way thus achieving such benefits as:

– higher quality of metrics – since the knowledge base (including good and bad practices) are shared easier and faster through the centralized metrics storage/team
– lower maintenance costs – since the centralized storage of metrics is optimized towards handling large quantities of data
– faster adoption of new metrics – since the metric team has the possibility to quickly assess the quality of metrics, has access to the relevant data sources and can reuse measurement systems between different parts of the company.

The method used in this paper is a case study where we define the theoretical framework – MaaS – a priori and use it to describe the measurement program at Ericsson (which is the unit of analysis). Our preposition is that by describing the measurement program using the MaaS conceptual framework we can identify new improvement areas – e.g. how to define value propositions for metrics.

The paper is structured as follows. Section 2 explains the main concepts and elements used in measurement-as-a-service. In section 3 we describe how we used MaaS to describe the measurement program at Ericsson. In section 4 we describe the main related work to our study and finally in section 5 we summarize the main message of the paper and outline the current research directions in this area.

2 Measurement-as-a-Service

MaaS is a measurement licensing and delivery model in which metrics are licensed on a subscription basis, centrally hosted, collected and delivered on demand. This concept is similar to the concept of Software-as-a-Service and Platform-as-a-Service. Figure 1 shows what MaaS consists of and how it relates to the core business processes (e.g. software development).

The main two types of actors in this context are the *Metrics team* and the *Stakeholder*. The metric team is responsible for the measurements – both process-wise (eliciting metrics, developing measurement systems, deploying information
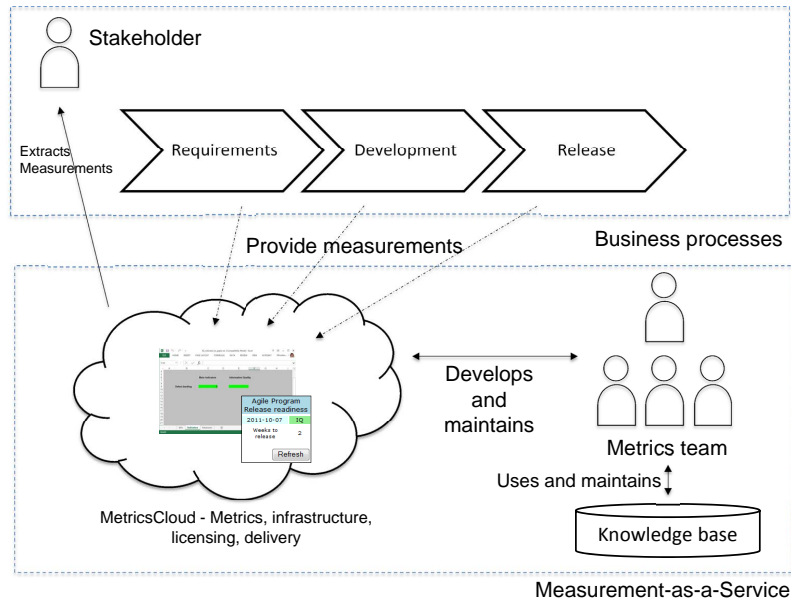
Fig. 1. Conceptual model of a measurement program

products) and competence-wise (assessing the quality of metrics and indicators, optimizing the number of metrics collected). The stakeholders are responsible for their business processes and/or products and use the measures to make decisions in their work [Sta12].

The metrics team is responsible for the development of measures and indicators based on the discussions with the stakeholders. The team has the responsibility for the long-term maintenance of the measurement knowledge base. The measurement knowledge base is the set of documented experiences and artifacts which have been proven to be useful for the organization, the set of best practices and the set of common pitfalls (e.g. measures which were found to be incorrect or leading to negative effects).

The metrics are naturally delivered as a product – *information product* according to ISO/IEC 15939. Examples of the information products can be MS Excel files, web pages with dashboards or MS Sidebar Gadgets. The delivery method for the information products can vary, but is usually similar to the SaaS – using the concept of cloud computing – e.g. [SM14].

The main benefit from organizing the measurement program as MaaS is the clear separation of competence in the organization – stakeholders focus on their business processes whereas the metrics team is the main point-of-contact for the measurement competence.

2.1 Central hosting

One of the key challenges in managing the measurement program is the ability to deliver the right metrics for the right stakeholders at the right time. The variability of stakeholders and their goals usually causes the number of measurement systems in the measurement program to grow over time. As the number of measurement systems grew in our collaborating organization and different dissemination patterns appeared (e.g. the distinction between public and local metrics), the company started to introduce an internal, cloud-based metrics dissemination system.

2.2 Collection and licensing

Although the majority of measurement systems within one organization can be available for all stakeholders, it is often practical to maintain a control over who uses which measurement system – license them. Since (according to ISO/IEC 15939) the measurement systems are dedicated for specific stakeholders, it is natural that the stakeholder's role in the company and the sensitivity of the information dictates the availability of the measurement system. The stakeholder can "license" the measurement system to be: (i) public (everyone can have access to the measurement system with read access right), (ii) local (private, only selected stakeholders can have access to the measurement system with read/write access), and (iii) shared (selected stakeholder can have access to the measurement system with read access). The metrics team manages the licensing – it could be a dedicated measurement support team or a set of roles spread across the company.

In MaaS, licensing is done per subscription. Stakeholders can subscribe to a specific kind of metric or indicator which they need for a particular purpose. The licensing should be time-limited in order to limit the number of unused metrics. It also allows the metric team to focus on the most important tasks at the moment and do not maintain unused metrics.

Collection, however, has a different pattern. Metrics should be collected (especially the base measures) even if they are not used by any stakeholders. The collected metrics can be used for visualizing trends for the stakeholders when they subscribe to the metrics and indicators. In this way the stakeholders have the incentive to use the MaaS supplier rather than to spend time on collecting the data themselves (in periods of time when the metrics is seemingly not useful).

The infrastructure and the automated execution environment form a measurement program together with the measurement systems, source files, raw data, databases and stakeholders. The measurement program is maintained by a measurement team which consists of designers and measurement agents. The metric team consists of the following roles: (i) designers – responsible for design, implementation and maintenance of the measurement systems, (ii) measurement agents – responsible for contacts with stakeholders to elicit information needs in the organization and keep the design of the existing indicators up-to-date, and

(iii) metric champions – responsible for identifying and introducing new metrics into the organization.

2.3 Delivered on demand

The delivery of the metrics should be done on demand – i.e. when the stakeholders who subscribe to a particular metric want to access the data. They could also be delivered automatically when new measurements are available (e.g. nightly after the automated measurement process has been executed).

The on-demand delivery does not require extra storage from the stakeholders' side, but utilizes the central storage. However, it makes the subscribed metrics available when needed (through the use of underlying cloud technologies such as MetricCloud).

2.4 Role of the metrics team

The supplier of MaaS is usually the metrics team at the company. The role of the team is then to identify the information needs of the organizations, identify measurements of importance, develop and provision measurement systems and manage the licensing of measurements in the organization. The metrics team realizes such tasks as the quality assurance of the measurement program, the functional development of the measurement program, operational and corrective maintenance and supporting the company with measurement competence.

This new responsibility of the metrics team extends the set of roles identified in the standard for measurement processes – ISO/IEC 15939 – by such roles as: measurement architect (responsible for the overall structure of the measurements, e.g. dependencies, links between information needs), measurement team leader (responsible for the coordination of efforts in the team, e.g. prioritizing assignments) and measurement account manager (responsible for contacts with specific unit, e.g. one product development unit). These roles, in particular the role of measurement account manager, are important for the continuity of the measurement program and its effectiveness in decision processes.

One of the main challenges for the metric team when operating in the MaaS context is the need to develop a *business model for measurement*. The team needs to describe metrics in terms of their value for the customers – *value proposition of the metric* which helps the team to "sell" the metrics to the stakeholders. Since the stakeholder's main focus in the main business of the company, this value propositions should link to the customer value that the company itself delivers.

2.5 Value propositions

The value proposition for each metric in the measurement program needs to include the goals which are important for the stakeholders, but it can contain common elements depending on the type of the stakeholder. The value proposition for a measurement should address such aspects as:

– Who should be the stakeholder for the measurement?
– How is the measurement linked to the goals of the stakeholder?
– Which information need of the stakeholder is fulfilled by the measurement?
– What value will the measurement bring if conducted?
– Which risks are related to conducting the measurement?

The template for a value proposition for a measurement should contain the elements described in table 1.

Table 1. Measurement value proposition for MaaS

Element	Characteristics	Purpose
Headline	Brief statement of the purpose of the measurement	Grabs attention
Sub-healine	A specific explanation of the measurement – what, for whom and why it is useful	Lists key benefits
Example	A simple example of what the measurement shows	Explains the context
Stakeholder	Role and mandate of the stakeholder	Links the measurement to the stakeholder
Value	Benefits from conducting the measurement	Value of the measurement
Risks	A list of risks of conducting the measurement, e.g. sub-optimizations	Describes the potential problems

An example of a value proposition for a release readiness indicator [SMP12] is presented in table 2.

Table 2. Measurement value proposition for MaaS

Element	Characteristics
Headline	Release readiness indicator
Sub-healine	Shows project managers how many weeks are needed to finish the product with a given quality
Example	When the project is finished with new feature development it shows how many weeks the testing and bug-fixing will take
Stakeholder	Release project manager
Value	Can decrease the cost of re-planning by as much as 30%
Risks	Releasing the product with minimum viable functionality

The value proposition provides a support for stakeholders in adopting the new measurement and allows to make a decision whether to buy a license for this type of service from the MaaS supplier.

3 Case study – using MaaS to describe the measurement program at Ericsson

In this paper we introduce the term MaaS, and in this section we use it to describe the way in which the measurement program at one of the units of Ericsson is organized. Ericsson AB (Ericsson) develops large software products for the mobile telecommunication network. The size of the organization during the study is several hundred engineers and the size of the projects is up to a few hundreds. Projects are increasingly often executed according to the principles of Agile software development and Lean production system. In this environment, various teams are responsible for larger parts of the process compared to traditional processes: design teams (cross-functional teams responsible for complete analysis, design, implementation, and testing of particular features of the product), network verification and integration testing, etc. The organization uses a number of measurement systems for controlling the software development project (per project) described above, a number of measurement systems to control the quality of products in field (per product) and a measurement system for monitoring the status of the organization at the top level. All measurement systems are developed using the in-house methods described in [SMN08], with the particular emphasis on models for design and deployment of measurement systems presented in [SM09c]. The needs of the organization evolved from metrics calculations and presentations (ca. 9 years before the writing of this paper), to using predictions, simulations, early warning systems and handling of vast quantities of data to steer organizations at different levels, and providing information from project and line. These needs are addressed by the action research projects conducted in the organization, since the 2006.

3.1 Measurement program at Ericsson

Measurement programs in industry are socio-technical systems where the technology interacts with stakeholders in order to support certain goals. Even though the term *measurement program* is defined in literature, the international standard ISO/IEC 15939:2007 (Systems and Software Engineering: Measurement process) introduces the concept of *measurement management system* which comprises both the measuring systems (e.g. instruments for data collection and visualization), the infrastructure where these operate, the knowledge bases on the use of measures and the stakeholders involved in the measurement process as conceptually shown in fig. 2.

   The central element of the measurement program is the set of measurement systems and information products. The *measurement systems* are dedicated software applications, designed for measuring quantities, addressing the stakeholder's information needs. The quantities are assembled (or combined) together in order to form indicators which, together with the analysis models, are packaged into information products.

   There are multiple solutions about how to realize measurement systems – for example using business intelligence tools and their reporting functionalities

or using simplistic MS Excel files (which is shown in fig. 2). The measurement systems combine the inputs from multiple measurement instruments (either directly of by querying databases) in order to calculate the indicators. The process is specified in the hierarchical measurement information model of the ISO/IEC 15939 standard.

The input to the measurement program is obtained by measuring properties of products, organizations (people) and processes. The measurement is often done by using measurement instruments (e.g. metrics tools) which quantify properties of one entity (e.g. source code of a program) into numbers (e.g. McCabe complexity number). These measurement instruments are often specialized for measuring properties of single entities of single types (e.g. complexity of the C code).

The output of the measurement program is a set of decisions taken in the organizations, the insights into the organizations' processes, products and projects and the early warnings of the coming problems and challenges. These are usually interconnected – e.g. insights can trigger decisions, decisions can require new insights when being implemented [Sta12].
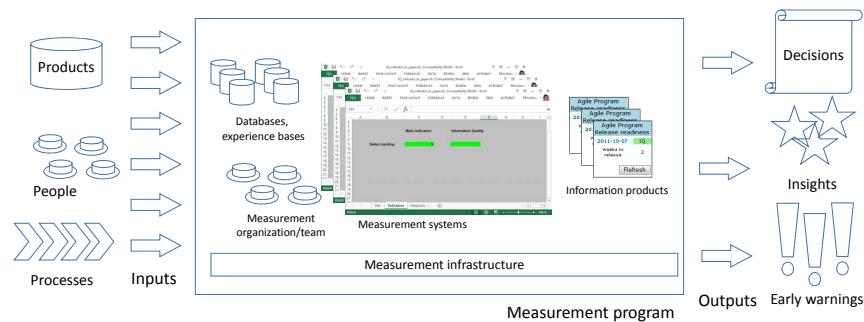

Fig. 2. Conceptual model of a measurement program

3.2 Information products

Delivering measurement information across organizations can be done in multiple ways. The concepts of information radiators [RS05], metric tools [FP98], business intelligence [EW07] or visual analytics [TC06] were coined for this purpose and each concept describes a specific kind of a measurement system. The work presented in this paper is compatible with these concepts as self-healing is important for all of them – the analyses can be reliable if the right data is available. In order to standardize the discussions and put self-healing in the context, we use the internationally adopted standard for developing measurement programs ISO/IEC 15939 (Software and Systems Engineering measurement processes) [OC07]. An alternative to ISO/IEC 15939 method for defining measures was presented by Chirinos et al. [CLB05], which is based on a meta-model

for measures proposed by authors created by combining certain aspects of GQM (Goal Question Metric, [VSBCR02]) into ISO/IEC 15939. In the case of Ericsson the information product is a measurement system.

A typical measurement system at Ericsson is built based on MS Excel and its scripting language VBA (Visual Basic for Applications) as presented in figure 3. The main worksheet of the MS Excel file (the grayed page at the top of the figure) contains indicators (green cells in the grayed page) if they are defined by the stakeholders, otherwise it contains values of measures. The indicators worksheet has the associated base and derived measures in other worksheets of the MS Excel file. These measures and indicators are calculated using VBA scripts (VBA for calculating measures and indicators) and VBA scripts for accessing the raw data from other measurement systems.
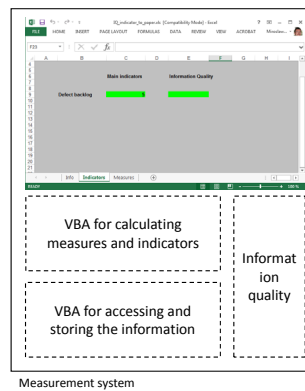

Fig. 3. Example of a measurement system

The basic control of the quality of the information is done by a separate set of VBA scripts (Information quality) and the result of the control is visualized as one of the indicators on the main page.

Such an architecture of measurement systems is aligned with the prescriptions of the standard [OC07] with the separation between base/derived measures and indicators, associated decision criteria and algorithms for data processing. This architecture is also scalable as it allows developing new measurement systems based on the existing ones (e.g. allowing to reuse existing derived measures in other measurement systems) yet providing the measurement systems towards dedicated stakeholders. Each stakeholder has his/her own measurement system fulfilling his/her information needs.

Another example of an information product at Ericsson is a dashboard presented in fig. 4. It shows the usage of a network in a laboratory environment and is dedicated for the project team to observe the status of their test network. For confidentiality reasons the names of the tested products are covered with greyed boxes.
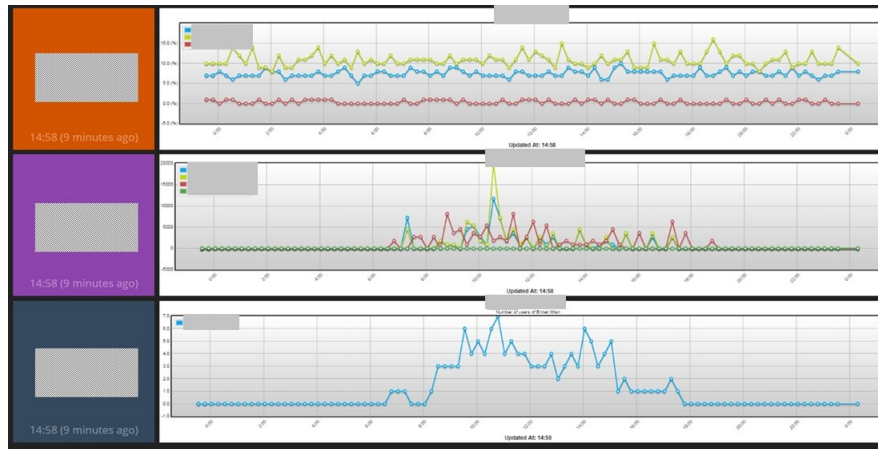
Fig. 4. An example of information radiator

The use of both types of information products differs as the first one is dedicated for decision support for particular stakeholders, is interactive and provides the possibility to access detailed data like trends. The dashboards are dedicated for spreading the information to larger populations (e.g. a project team) and is supposed to contain succinct information that provides enough details so that the users do not need to interact with the dashboard.

3.3 On-demand delivery – MS Sidebar gadgets

An example of the on-demand delivery is the use of MS Sidebar gadgets. A gadget is used as a placeholder for the content on the stakeholders' computer, but the information itself is served through the metric infrastructure [SMN09]. An example of such a gadget is presented in fig. 5.
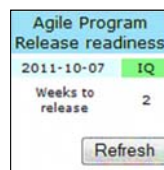

Fig. 5. MS Sidebar gadget – an example of on-demand delivery

The gadget presents the number of weeks to release as defined in our earlier work [SMP12] and presents the indicator only. Once clicked the entire measurement system in MS Excel is fetched from the server and is presented to the stakeholder (on-demand delivery). This kind of on-demand delivery combined

with continuous data collection could help to remedy the problems with missing data in software processes – e.g. [AN93].

3.4 Centralized storage – MetricsCloud

MetricsCloud is an infrastructure for disseminating measurement systems used at Ericsson [SM14]. MetricsCloud addresses such needs of the organization's stakeholders as (s-i) dissemination of self-managed measurement systems,(s-ii) possibility to share measurement systems, and (s-iii) obtaining simple measurement execution infrastructure. MetricsCloud also provides benefits to the metric team: (m-i) reducing the need to create "simple" measurement systems - now done by stakeholders, (m-ii) applying identity-based security, and (m-iii) reducing the need to constantly keep-alive the web-server with all measurement systems.

The dissemination of metrics based on MetricsCloud separates the concerns of information delivery and execution/storage of information. This separation of concerns is done by designing cloud systems based on layers according to the principles defined by Pallis et al. [PAL10]. Pallis et al. identifies such layers in a cloud-based system in general – e.g platform, infrastructure. In this paper we instantiate three of these layers based on the division of responsibility (in the organization): (i) Information product delivery, (ii) Execution and information quality, and (iii) Storage and access as presented in fig. 6.

Fig. 6. Layers in cloud infrastructure

The top layer contains measurement systems managed individually by stakeholders of measurement systems who need access to information (addressing the needs of s-i, s-ii and m-i). The mid-layer is the layer of execution and update of measurement systems and is managed by the dedicated metric team. The stakeholders of the measurement systems do not need to be concerned about the execution of public measurement systems, but are notified if the measurement systems are not updated (e.g. by information quality indicators [SM09a]). Finally the lowest layer is the standard IT infrastructure of the company with network file servers, web servers and client programs which is managed by the IT department of the company.

3.5 Evolution

The existing measurement program at the studied unit evolved from their decentralized set-up to MaaS measurement programs in a number of steps summarized in table 3

Table 3. Evolution of measurement programs to MaaS

Step	Characteristics
Standardization of measurement systems	Standardized measurement systems based on common tools – e.g. MS Excel, Tableau, QlikView
Common support	Established metric team with the dedicated roles to support the organization
Centralized distribution	Centralization of the distribution of the metrics – e.g. MetricCloud
Licensing	Development of a business model for "selling" metrics to different stakeholders and organizations within the company
Research	Identification and development of new metrics, combined with the business model for "selling" measurements

This evolution helped the organization to centralize the measurements and "outsource" them internally or externally to the metric team. The metric team has the opportunity to develop a business model where the value of the metrics for the stakeholders is the main interest. This focus on the value helps to emphasize the metrics which bring more value to the company and de-prioritize the metrics which are not that important.

4 Related Work

One of the aspects important in the use of MaaS internally at companies is the understanding how information product spread – i.e. the internal communication channels and the reusability of metric. One of the works in this area is the work of Atkins et al. [AMVP03] which presents the models for reusability of metrics. Our work complements the reusability aspects by providing the value proposition.

Another work in this area is the work of Jorgensen et al. [Jor99]. As their work shows, this is not an easy task due to the potential different definitions of measures. Jorgensen shows contrasting definitions of measures if quality is defined as "a set of quality factors", "user satisfaction", and "software quality related to errors". Our research recognizes the needs for viewing the same aspects (e.g. quality) from different perspectives - depending on the stakeholder. These needs are also recognized by the measurement team which we collaborated with.

The delivery method for metrics – MetricCloud – have been validated at Ericsson in our previous work. This validation is aligned with the work of Pawluk

et al [PSS$^+$12] who described the process of introducing a new cloud solution to a large enterprise. The purpose of the cloud is similar to ours and we use their work when designing our cloud system. The current cloud system is an evolution of the previous work on ensuring information reliability done together with Ericsson [SM09a]. In this work we address the problems of ensuring that information is available throughout the enterprise and its understanding [KS02, SKT05, MS10, MST12].

Yoon et al. [YOL13] showed how to establish security into cloud computing. The security of MetricsCloud is based on similar principles but is a simplification of the security policies. All security is based on the enterprise log-in. The licensing model of the measurements, important for MaaS, need the kind of security described by Yoon et al.

Another approach was presented by Zhang et al [ZZ09] and their CCOA framework. Although a very elaborate framework could be used in our solution we preferred to use a simple approach and focus on the ease-of-use. It is the ease-of-use and performance which are important for similar cloud systems as described by Gong et al [GLZ$^+$10].

Farooq et al. [FKDW06] presented an approach for structuring the measurement process (ISO/IEC 15939 based) using web services in order to increase scaleability and reuse of metrics. We complement their approach by adding the explicit role of the metrics team, licensing and value propositions for software metrics.

Sakamoto et al. [SMSN13] have developed a tool for mining software metrics and storing them in a web service environment. Their study is a good complement to our work as it addresses the question of metrics acquisition from large software repositories.

5 Conclusions

Modern large software development organizations focus on delivering customer value and often adopt decentralized software development models such as Agile. In these models various units of the organizations can work independently and communicate often. Measures, indicators and Key Performance Indicators are examples of communication means. However, the challenge in such organizations is to manage these means – e.g. keeping them consistent, secure and available on-demand. In this paper we addressed the problem of how to manage this in an efficient way by using a newly introduced concept – Measurement-as-a-Service, MaaS.

MaaS is a measurement licensing and delivery model in which metrics are licensed on a subscription basis, centrally hosted, collected and delivered on demand. This concept is similar to the concept of Software-as-a-Service and Platform-as-a-Service. We introduced this term in this paper and we used it to describe the measurement program at Ericsson – one of our industrial partners. We have shown that Measurement-as-a-Service is targeted towards improving the internal and external management of measurement programs. We showed

that using modern, yet simplistic, cloud-based dissemination systems allows to develop new business models for measurement distribution. This new business model allows the stakeholders for measurement systems to focus on their core business activities while leaving the core metric competence to a dedicated entity.

The benefits of using Measurement-as-a-Service help the company to become more effective in their decision processes and allows them to focus on delivering customer value, at the same time having fact-based decisions. The benefits from using MaaS include such aspects as:

– higher quality of metrics – since the knowledge base (including good and bad practices) are shared easier and faster through the centralized metrics storage/team
– lower maintenance costs – since the centralized storage of metrics is optimized towards handling large quantities of data
– faster adoption of new metrics – since the metric team has the possibility to quickly assess the quality of metrics, has access to the relevant data sources and can reuse measurement systems between different parts of the company.

In our further work we intend to explore the notion of value proposition of metrics by studying the value propositions used in our industrial partner. We plan to develop a generic model for describing metrics and indicators in a business-like manner in order to reduce the number of "wrong" metrics collected in industry today.

6 Acknowledgements

References

[AMVP03] Kenneth L. Atkins, Bredt D. Martin, Joseph M. Vellinga, and Rick A. Price. Stardust: implementing a new manage-to-budget paradigm. *Acta Astronautica*, 52(2-6):87–97, 2003. TY - JOUR.

[AN93]   Alain Abran and Hong Nguyenkim. Measurement of the maintenance process from a demand-based perspective. *Journal of Software Maintenance: Research and Practice*, 5(2):63–90, 1993.

[CLB05]   Ledis Chirinos, Francisca Losavio, and Jørgen Bøegh. Characterizing a data model for software measurement. *Journal of Systems and Software*, 74(2):207–226, 2005.

[EW07]   Mohamed Elbashir and Steve Williams. Bi impact: The assimilation of business intelligence into core business processes. *Business Intelligence Journal*, 12(4):45, 2007.

[FKDW06]   Ayaz Farooq, Steffen Kernchen, Reiner R Dumke, and Cornelius Wille. Web services based measurement for it quality assurance. In *Proceedings of the International Conference on Software Process and Product Measurement (MENSURA 2006)*, pages 241–251, 2006.

[FP98]   Norman E Fenton and Shari Lawrence Pfleeger. *Software metrics: a rigorous and practical approach.* PWS Publishing Co., 1998.

[GLZ$^+$10] Chunye Gong, Jie Liu, Qiang Zhang, Haitao Chen, and Zhenghu Gong. The characteristics of cloud computing. In *Parallel Processing Workshops (ICPPW), 2010 39th International Conference on*, pages 275–279. IEEE, 2010.

[JA97]   J-P Jacquet and Alain Abran. From software metrics to software measurement methods: a process model. In *Third IEEE International Software Engineering Standards Symposium and Forum*, pages 128–135. IEEE, 1997.

[Jor99]   M. Jorgensen. Software quality measurement. *Advances in Engineering Software*, 30(12):907–912, 1999.

[KS02]   Ludwik Kuzniarz and Miroslaw Staron. On practical usage of stereotypes in uml-based software development. In *Forum on Design and Specification Languages*, 2002.

[MS10]   Niklas Mellegård and Miroslaw Staron. Characterizing model usage in embedded software engineering: a case study. In *Proceedings of the Fourth European Conference on Software Architecture: Companion Volume*, pages 245–252. ACM, 2010.

[MST12]   Niklas Mellegard, Miroslaw Staron, and Fredrik Torner. A light-weight defect classification scheme for embedded automotive software and its initial evaluation. In *Software Reliability Engineering (ISSRE), 2012 IEEE 23rd International Symposium on*, pages 261–270. IEEE, 2012.

[OC07]   International Standard Organization and International Electrotechnical Commission. Software and systems engineering, software measurement process. Technical report, ISO/IEC, 2007.

[PAL10]   George PALLIS. Cloud computing: The new frontier of internet computing. *IEEE internet computing*, 14(5):70–73, 2010.

[PSS$^+$12] Przemyslaw Pawluk, Bradley Simmons, Michael Smit, Marin Litoiu, and Serge Mankovski. Introducing STRATOS: A cloud broker service. In *IEEE CLOUD*, pages 891–898, 2012.

[RS05]   Hugh Robinson and Helen Sharp. Organisational culture and xp: three case studies. 2005.

[SKT05]   Miroslaw Staron, Ludwik Kuzniarz, and Christian Thurn. An empirical assessment of using stereotypes to improve reading techniques in software inspections. In *ACM SIGSOFT Software Engineering Notes*, volume 30, pages 1–7. ACM, 2005.

[SM09a]   Miroslaw Staron and Wilhelm Meding. Ensuring reliability of information provided by measurement systems. In *Software Process and Product Measurement*, pages 1–16. Springer, 2009.

[SM09b]   Miroslaw Staron and Wilhelm Meding. Transparent measures: cost-efficient measurement processes in SE. 1:1–10, 2009.

[SM09c]   Miroslaw Staron and Wilhelm Meding. Using models to develop measurement systems: A method and its industrial use. 5891:212–226, 2009.

[SM14]   Miroslaw Staron and Wilhelm Meding. Metricscloud: Scaling-up metrics dissemination in large organizations. *Advances in Software Engineering*, 2014, 2014.

[SMKN10]   M. Staron, W. Meding, G. Karlsson, and C. Nilsson. Developing measurement systems: an industrial case study. *Journal of Software Maintenance and Evolution: Research and Practice*, pages n/a–n/a, 2010.

[SMN08]   Miroslaw Staron, Wilhelm Meding, and Christer Nilsson. A framework for developing measurement systems and its industrial evaluation. *Information and Software Technology*, 51(4):721–737, 2008.

[SMN09]   Miroslaw Staron, Wilhelm Meding, and Christer Nilsson. A framework for developing measurement systems and its industrial evaluation. *Information and Software Technology*, 51(4):721–737, 2009.

[SMP12]   Miroslaw Staron, Wilhelm Meding, and Klas Palm. Release readiness indicator for mature agile and lean software development projects. In *Agile Processes in Software Engineering and Extreme Programming*, pages 93–107. Springer, 2012.

[SMSN13]   Yuta Sakamoto, Shinichi Matsumoto, Sachio Saiki, and Mitsutoshi Nakamura. Visualizing software metrics with service-oriented mining software repository for reviewing personal process. In *Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD), 2013 14th ACIS International Conference on*, pages 549–554. IEEE, 2013.

[Sta12]   Miroslaw Staron. Critical role of measures in decision processes: Managerial and technical measures in the context of large software development organizations. *Information and Software Technology*, 54(8):887–899, 2012.

[TC06]   James J Thomas and Kristin A Cook. A visual analytics agenda. *Computer Graphics and Applications, IEEE*, 26(1):10–13, 2006.

[VSBCR02] Rini Van Solingen, Vic Basili, Gianluigi Caldiera, and H Dieter Rombach. Goal question metric (gqm) approach. *Encyclopedia of Software Engineering*, 2002.

[YOL13]   Young Bae Yoon, Junseok Oh, and Bong Gyou Lee. The establishment of security strategies for introducing cloud computing. *KSII Transactions on Internet and Information Systems (TIIS)*, 7(4):860–877, 2013.

[ZZ09]   Liang-Jie Zhang and Qun Zhou. CCOA: Cloud computing open architecture. In *Web Services, 2009. ICWS 2009. IEEE International Conference on*, pages 607–616. Ieee, 2009.