

Research Article

MetricsCloud: Scaling-Up Metrics Dissemination in Large Organizations

Mirosław Staron¹ and Wilhelm Meding²

¹Computer Science and Engineering, University of Gothenburg, Förskningsgången 6, 412 96 Gothenburg, Sweden

²Ericsson AB, 164 83 Stockholm, Sweden

Correspondence should be addressed to Mirosław Staron; miroslaw.staron@gu.se

Received 10 September 2014; Revised 17 November 2014; Accepted 18 November 2014; Published 10 December 2014

Academic Editor: Phillip A. Laplante

Copyright © 2014 M. Staron and W. Meding. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The evolving software development practices in modern software development companies often bring in more empowerment to software development teams. The empowered teams change the way in which software products and projects are measured and how the measures are communicated. In this paper we address the problem of dissemination of measurement information by designing a measurement infrastructure in a cloud environment. The described cloud system (MetricsCloud) utilizes file-sharing as the underlying mechanism to disseminate the measurement information at the company. Based on the types of measurement systems identified in this paper MetricsCloud realizes a set of synchronization strategies to fulfill the needs specific for these types. The results from migrating traditional, web-based, folder-sharing distribution of measurement systems to the cloud show that this type of measurement dissemination is flexible, reliable, and easy to use.

1. Introduction

Managing software products and software development projects in large organizations has evolved over the past decade, from the central, top-down planning of waterfall processes to distributed monitoring of empowered agile teams [1]. The practices of software management have evolved as well from following the plans to adjusting the plans based on customer needs [2]. In modern software development processes it is the software development team that is responsible for planning their work in order to deliver customer value (e.g., new features) in a (more-or-less) continuous manner [3, 4]. From the perspective of software management in general and software measurement in particular this change requires a change in how measuring software products and projects is conducted and disseminated in enterprises. Using business intelligence tools [5, 6] provides managers and product/program leaders with the insight into the organization but is usually burdened with high maintenance costs [7]. The business intelligence tools are often complemented with the so-called information radiators which are designed to spread the information

throughout enterprises [8]. However, both manners are mainly used to communicate vertically in the hierarchy of the enterprise (teams-management and management-teams). In the context of empowered teams this vertical communication needs to be complemented with a horizontal dissemination of measurement information between the teams without causing information chaos or extensive information noise. In this paper we address this need by designing an infrastructure based on the principles of IaaS (Infrastructure as a Service [9]) where empowered teams and individual stakeholders can share their measurement information without creating information chaos. The infrastructure has evolved from an extensive measurement program at one of the development units at Ericsson and addresses its needs for scalability, distribution, reliability, and low maintenance cost [10]. The infrastructure is named MetricsCloud. MetricsCloud is based on file synchronization in a controlled manner, specific to the purpose—vertical or horizontal dissemination and public or private access. The infrastructure uses a central storage for measurement systems and a set of local clients to distribute, visualize, and update the measurement systems. At Ericsson

this infrastructure uses network storage, web pages, and execution engines based on our previous work [11].

In this paper we address the following research question: *How to increase dissemination of software metrics using cloud technologies?*

We describe how MetricsCloud infrastructure is designed and we provide lessons learned for the companies willing to use cloud solutions internally—both to increase the measurement dissemination and as a means of internal education on how IaaS cloud systems can be designed. We also evaluate MetricsCloud by introducing it for a selected group of measurement systems and interviewing them before and after the introduction of MetricsCloud.

Our results show that cloud-based dissemination increases the reliability of the entire measurement infrastructure (replication of information) and seamless availability of the measurement infrastructure (scaled-down distributed version of measurement execution) but increases the need of security control and monitoring of measurement systems (for private and shared measurement systems).

The paper is structured as follows. Section 2 presents research related to our work. Section 3 shows how measurement systems are designed and used at Ericsson. Section 4 describes the cloud environment used to increase the dissemination of metrics at the company. Section 5 shows how the cloud design is realized at Ericsson. Section 6 summarizes lessons learned from developing the cloud environment for a software development organization. Finally Section 7 outlines the conclusions and further work.

2. Related Work

Pawluk et al. [12] 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 [11]. In this work we address the problems of ensuring that information is available throughout the enterprise.

Yoon et al. [14] showed how to establish security in 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.

Vecchiola et al. [15] presented a solution based on NET for implementing cloud systems. Despite the similar technology their cloud solution does not address the issues specific to measurement programs in the enterprise. In particular it does not address the need of combining IaaS with the execution environment (SaaS). This is addressed in the work of Liu et al. [16].

Another approach was presented by Zhang and Zhou [17] 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 that are important for similar cloud systems as described by Gong et al. [18].

Security is an important aspect which has been recognized by Pearson et al. [19]. The MetricsCloud infrastructure implements the most important security aspects based on the enterprise access control, which is aligned with Person et al.'s approach.

3. Measurement Systems at Ericsson

The organization within Ericsson, which we worked closely with, develops large products for the telecommunication networks. The size of the organization 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 referred to as streamline development (SD) within Ericsson [20]. In this environment various disciplines 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, and so forth.

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.

One example of a measurement system for the monitoring of the status of the organization is the measurement system monitoring feature burndown. It is updated daily and is used by project management to control the status of the development. It is implemented using MS Excel and VBA (Visual Basic for Applications) scripts. Another example is a measurement system for monitoring the velocity of integrating new features to the main product code base.

All measurement systems are developed using the in-house methods described in [10, 21], with the particular emphasis on models for design and deployment of measurement systems presented in [22, 23]. The needs of the organization evolved from metric calculations and presentations (ca. 8 years before the writing of this paper) to using predictions, simulations, early warning systems, and handling of vast quantities of data to manage organizations at different levels and providing information from teams to management. These needs are addressed by a number of action research projects which we conducted in the organization, since 2006.

The studied organization uses and maintains over 400 different measurement systems (over 4000 MS Excel files) and a large number of persons are involved in the decision processes based on the status of measures. A graph with the size of measurement systems at Ericsson is presented in Figure 1.

As the figure shows there are a significant number of files with over 1000 lines of VBA code, which indicates the complexity of the updates and dissemination of the measurement systems.

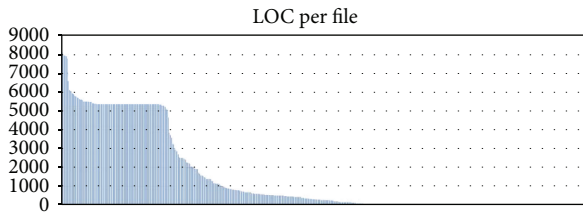


FIGURE 1: Size of measurement systems measured in number of lines of VBA code.

The measurement systems are at all levels of the organization and include measures for, for example,

- (i) measuring reliability of network products in operation for the manager of the product management organization (Figure 3); example measures in this measurement system are
 - (a) product downtime per month in minutes,
 - (b) number of nodes in operation;
- (ii) measuring project status and progress for project managers and team leaders who need to have daily updated information about such areas as requirements coverage in the project, test progress, and costs; example measures in this measurement system are
 - (a) number of test cases executed during the current week,
 - (b) cost of the project up till the current date,
 - (c) development efficiency,
 - (d) in-development product quality;
- (iii) measuring postrelease defect inflow for product managers who need to have weekly and monthly reports about the number of defects reported from products in field; examples of measures are
 - (a) number of defects reported from field operation of a product during the last month,
 - (b) predicted number of defects per week;
- (iv) summarizing status from several projects for department managers who need to have an overview of the status of all projects conducted in the organization, for example, number of projects with all indicators green.

The information is processed as a flow, according to the pipes-and-filters architectural style as presented in Figure 2.

To keep the measurement systems updated daily (or several times a day) the metrics team at Ericsson has developed its own infrastructure based on a set of simple scripts. The scripts export raw data from databases, execute measurement systems, and present the indicators for the stakeholders. This is done (usually) during night and takes a few hours (for all measurement systems).

There are multiple advantages of this type of solution; for example,

- (i) using MS Excel files makes the measurement systems available to all stakeholders (MS Office is a standard software package installed on every computer);
- (ii) centralized storage is backed up and kept up to date by a dedicated metric team;
- (iii) using MS Sidebar gadgets and web pages is a simple and effective way to spread the information across the company.

There are, however, limitations with this way of working. One of the main limitations is that the centralization requires all measurement systems to be maintained by the metric team. Some stakeholders would like to maintain their own measurement systems and only disseminate them using the measurement infrastructure—similar to Amazon’s cloud solution [24].

4. Cloud Systems

Cloud computing [25, 26] introduces new paradigms in information processing. It implements a modern vision of “sharing” resources and infrastructure between organizations, which is the focus of this paper (there exist other uses, for example, information processing. However, these are not part of this paper). Cloud systems are usually provided by commercial vendors which have the right competence and resources to manage cloud systems whereas “user organizations” can focus on their primary business objectives.

4.1. Cloud-Based Dissemination Patterns. In general many cloud-based *information dissemination systems* use Infrastructure as a Service (IaaS) where MetricsCloud (described in Section 5) is the infrastructure to store, update, and share measurement systems. This approach is described by Armbrust et al. [27].

One of the patterns is a simple file-sharing and one of the widely spread general-purpose *file-sharing systems* based on cloud is Dropbox [28] which provides users with a rudimentary mechanism of synchronizing files across platforms and sharing the files with other users. Systems like Dropbox pose no constraints on file types and leave it up to the user to design sharing patterns. Dropbox’s synchronization mechanisms are market leaders with batched synchronization and advanced clean-up mechanisms [29].

Another pattern is focusing on the information to be shared and is described by Rosenthal et al. [30] who present a cloud system for biomedical research which uses *information sharing* as the main objective regardless of its form (e.g., files or data). The pattern distinguishes two elements of a cloud-based dissemination of information—infrastructure/service providers and research consortium management.

These patterns are similar to web-based information sharing but have an advantage of replication of information and seamless synchronization. Figure 4 presents a comparison between the dissemination of information based on web-systems and cloud-based dissemination.

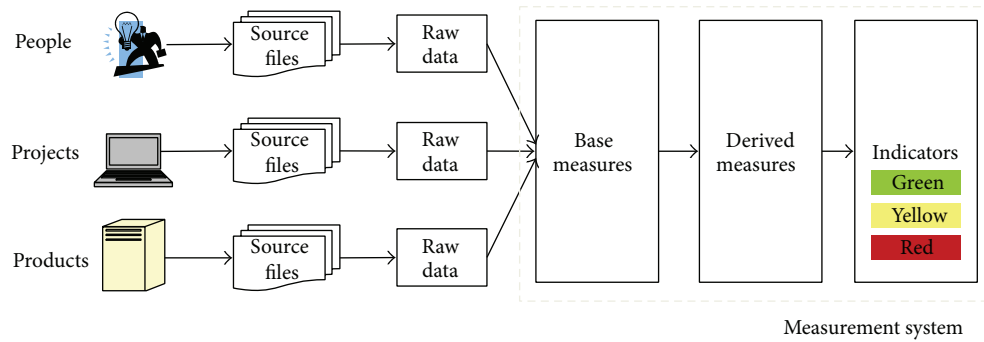


FIGURE 2: Information flow in a measurement system.

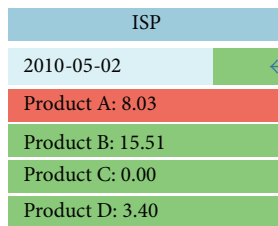


FIGURE 3: MS Sidebar gadget for visualizing product availability at Ericsson.

The major differences include (as mentioned before) the separation of concerns—information provision and execution/storage of information. The stakeholders of the measurement systems have access to their information on their clients (either computers or mobile devices) without the need to remember how to access the information over the internet (e.g., the URL of the server or a web page).

5. MetricsCloud

MetricsCloud addresses the needs of the organization for the stakeholders (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. In the current solution described in Section 3 the stakeholder relied on the metrics team to develop, update, and disseminate the measurement systems to all parties necessary. This could cause the metric team to become a bottleneck of dissemination of measurement systems in a large organization or introduce the necessary costs. One of the intentions with MetricsCloud was to reduce the risk of the bottlenecks and/or unnecessary costs.

As shown in Figure 4 the dissemination separates the concerns of information provision and execution/storage of information. This separation of concerns is done by designing cloud systems based on layers—Pallis [25] identifies such

layers in a cloud-based system in general, for example, platform, infrastructure. In this paper we instantiate three of these layers based on the division of responsibility (in the organization): (i) information product provisioning, (ii) execution and information quality, and (iii) storage and access as presented in Figure 5.

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 midlayer is the layer of execution and update of measurement systems and is managed by the dedicated metric team. The stakeholders of the measurement systems are not concerned with execution of public measurement systems but are notified if the measurement systems are not updated (e.g., by information quality indicators [11])—fulfilling the needs of s-iii, m-ii, and m-iii. 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.

5.1. Architecture. The architecture of the cloud-based dissemination is presented in Figure 6. The client is a program which provides the access and synchronization of measurement systems between the cloud (storage) and the client computers. Gadgets and web pages are used to provide access to the main server with the measurement systems [31].

The cloud architecture uses the following servers:

- (1) storage server—provides network access to all measurement systems;
- (2) execution server—updates measurement systems. The connection between the servers and the clients is presented in Section 5.2.

Each measurement system, which is part of either the public or the private part, is an MS Excel file which processes data from various sources as shown in Figure 2.

(1) Execution Environment. The measurement systems can be updated manually or automatically and both on the client and server side. The manual update of a measurement system is done by the stakeholder and then the MetricsCloud is used to disseminate the results of the stakeholder’s measurement process. They are synchronized from the client to the server.

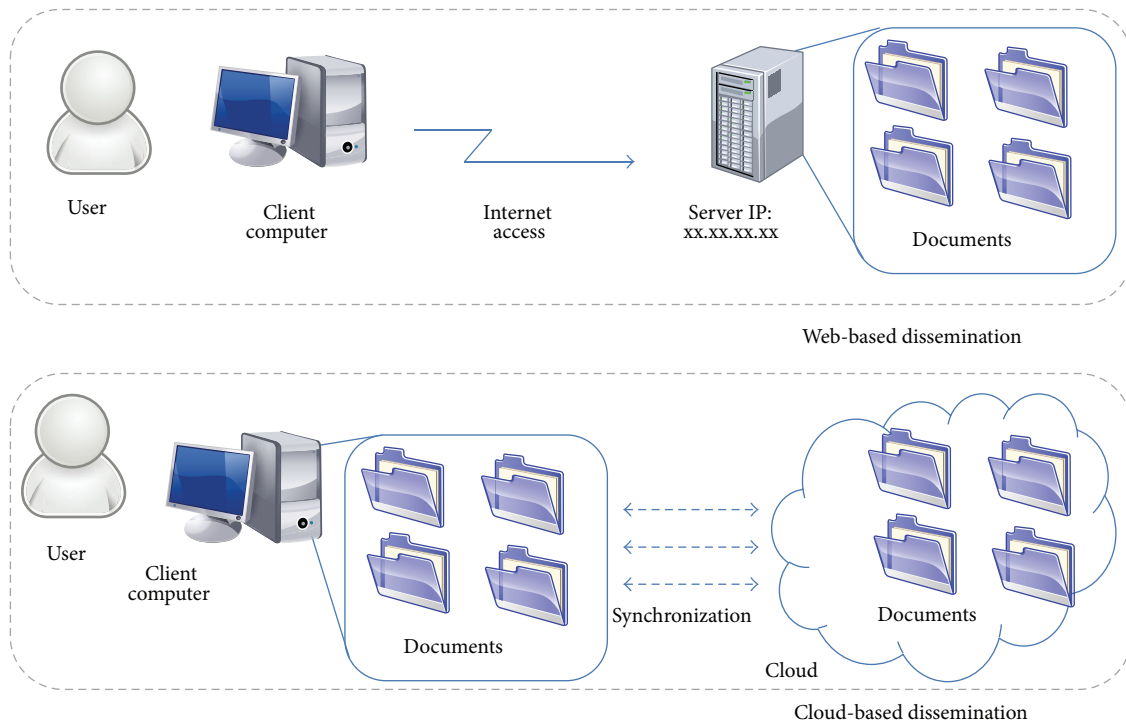


FIGURE 4: Web-based and cloud-based dissemination patterns.

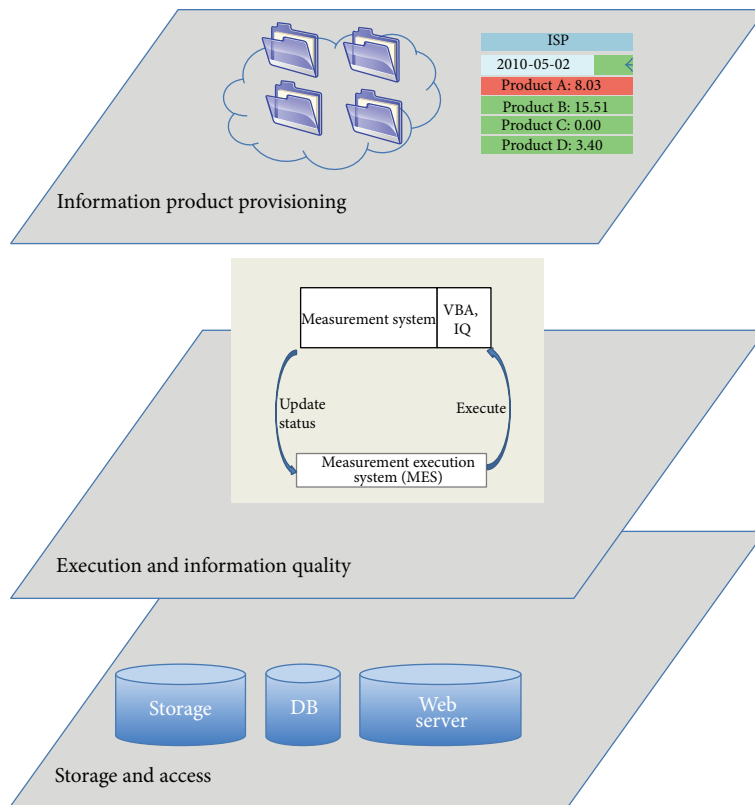


FIGURE 5: Layers in cloud infrastructure.

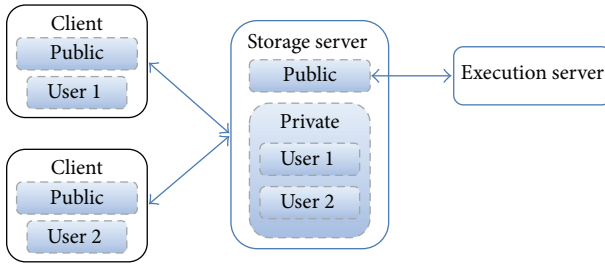


FIGURE 6: Cloud infrastructure—execution and storage servers and clients.

However, it is the automatically updated measurement systems that are more interesting. These need to be updated and synchronized correctly. This is done by a dedicated program—*measurement execution system, MES*—shown in Figure 7.

Every public measurement system is executed by MES and synchronized to the clients. However, the MetricsCloud client has a built-in simplified version of MES (i.e., excluded mechanisms for controlling the reliability of the measurement program infrastructure). The execution system executes the private measurement systems if any of the conditions is fulfilled:

- (i) measurement system is an MS Excel file with a macronamed “start_here”;
- (ii) measurement system is an executable file (.exe);
- (iii) measurement system is a Ruby script and a Ruby interpreter is available on the client machine.

MES is fail-safe and protects the user’s computer system from potentially defective measurement systems. It uses multithreaded execution and monitors problems with the “execution thread”—if needed it terminates the thread.

(2) *Size*. MetricsCloud is realized in C# and consists of ca. 1000 lines of code of the client. For the identification of measurement systems in the MetricsCloud we use such attributes as (i) file name, (ii) stakeholder company identity, or (iii) modification time.

(3) *Security*. Security in the MetricsCloud uses enterprise identities of stakeholders. Each MetricsCloud client discovers the corporate ID of the stakeholders and uses it when communicating with the storage server. The storage server steers which users have access to which folders. The MetricsCloud client synchronizes only those measurement systems which are (i) public, (ii) private for the stakeholder, and (iii) shared with the stakeholder.

(4) *Energy Optimization*. One could observe that it is possible to save the energy as the information is replicated to multiple machines. When the measurement systems are updated the execution server can be put to sleep and only the access to the updated files can be retained.

5.2. *Operation of MetricsCloud*. Based on our previous work on measurement systems [21] we have identified a number of types of measurement systems.

- (1) **Public**: these measurement systems are spread across the company. Everyone can access these measurement systems designed by designated stakeholders (e.g., project managers, product managers). An example of such measurement system is status of the total defect backlog in the project—information which should be spread in the organization.
- (2) **Private**: these measurement systems are designed by individual stakeholder and managed by them; that is, updates are done locally. The measurement systems are visible only to the users who manage them. An example of such measurement system is a measurement system for defect backlog for a designer; the measurement system may include its “own” information about the status of the repair of the defect.
- (3) **Shared**: these measurement systems are private but the owner has shared it with another user. An example is defect backlog for a team—this information should be spread within the team.
- (4) **Locally updated public**: these measurement systems are public, but they are updated by a single user. The user can be a stakeholder responsible for inputting data to the measurement system. For other users the measurement system is visible as public (like the 1st item). An example is the annotated feature backlog—teams update their information locally, but it is spread for the entire project.
- (5) **Centrally updated private**: these measurement systems are private for one stakeholder but are maintained by the central storage and execution server. These can be dedicated measurement systems with company-sensitive information. An example can be measurement systems with budget information which is dedicated to certain group of stakeholders within the organization.

Each of the above measurement systems requires different management by the MetricsCloud. These management strategies are based on the following synchronization principles (see Figure 8).

- (1) **Public**: these measurement systems are always updated from the storage server to the client.
- (2) **Private**: these measurement systems are always updated from the client to the storage server.
- (3) **Shared**: these measurement systems are always updated from the owner to dedicated clients.
- (4) **Locally updated public**: these measurement systems are updated from the client to the storage server.
- (5) **Centrally updated private**: these measurement systems are updated from the storage server to the dedicated client.

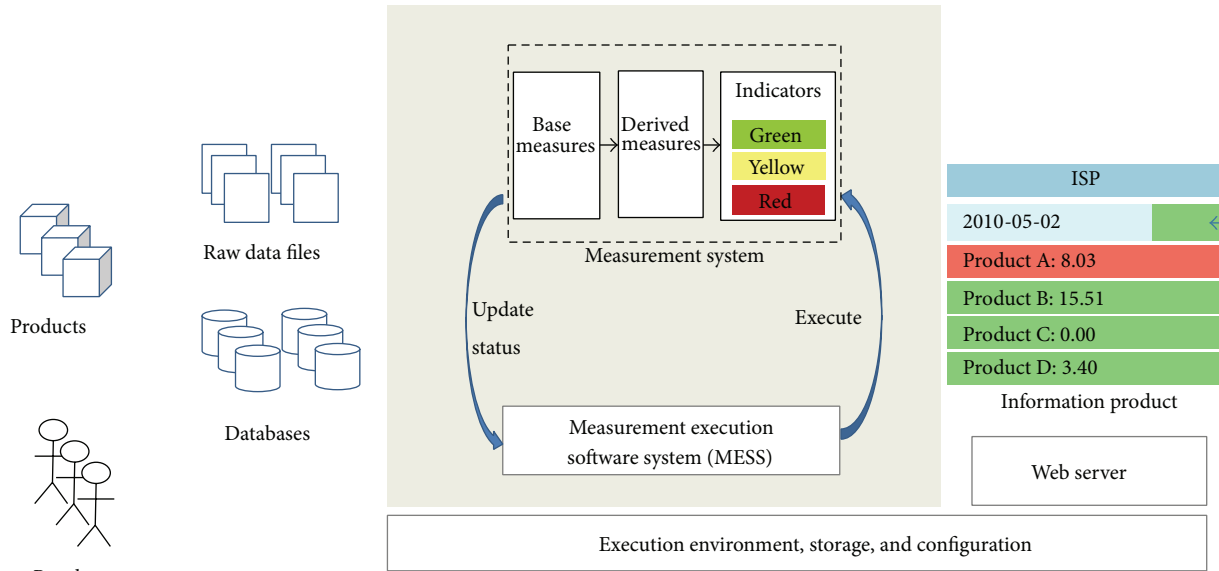


FIGURE 7: Measurement execution system.

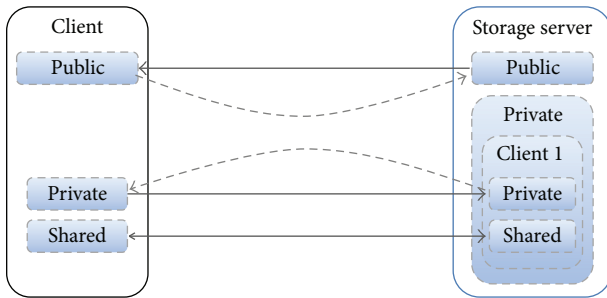


FIGURE 8: Synchronization of measurement systems.

In Figure 8 measurement systems of types 1–3 are synchronized along the solid lines using the pull mechanisms. It is the MetricsCloud client that is responsible for the synchronization. The measurement systems of types 4 and 5 are synchronized along the dotted lines. The dotted lines mean that these measurement systems have a file attribute set—offline. The MetricsCloud client recognizes this attribute and based on the location of the file makes the appropriate synchronization.

5.3. *Mapping Types to Existing Solutions.* The types of the measurement systems identified in this section can be found in other solutions (e.g., business intelligence tools) and support different types of communication in the company (as outlined in the Introduction). Table 1 maps the types to communication and other solutions.

However, these mechanisms are not integrated in the other solutions which can hinder the stakeholders (and thus the organizations) from efficient dissemination. Locally used metric tools require manual intervention and distribution lists to spread the information. Shared measurement systems

TABLE 1: Mapping of types of measurement systems to communication types and example of existing solutions.

Type of MS	Communication type	Examples of existing solutions
Public	Manag. <—> teams	Business intelligence
Public	Manag. <—> manag.	Business intelligence, scorecards
Private	Individual	MS Excel, metric tools [13]
Shared	Team <—> team	Business intelligence
Manually updated	N/A	MS Excel

for teams require mechanisms for controlling the quality of the information and maintenance of the common storage (usually a website).

These limitations are addressed by the MetricsCloud.

6. Evaluation

We evaluated the work by conducting interviews with the members of the metric team at Ericsson.

6.1. *Design of Evaluation.* Before introducing MetricsCloud we interviewed members of the metrics team and stakeholders of measurement systems at Ericsson. We asked questions regarding the following.

- (i) How do you use the current infrastructure for disseminating measurement systems?
- (ii) What are the main positive/negative aspects of the current measurement infrastructure?

- (iii) How do these positive/negative aspects affect your daily work?
- (iv) Which improvements would you recommend to the measurement program based on your experience with it?
- (v) What are the main limitations to the dissemination of measurement systems which stem from the current infrastructure?

These aspects allowed us to understand what the limitations of the infrastructure were both before introducing MetricsCloud into the organization and during evaluation after the introduction.

6.2. Results

(1) *Studying Growth of Measurement Systems.* As the first step in the evaluation we collected the statistics about the trends in indicators for one (out of many) software development program at Ericsson. The program develops a complex telecom product which is used for mobile data traffic communications. The trend in the number of measures and indicators for this program is presented in Figure 9.

In the figure we also added the major drivers for both the decreasing and the increasing trends. Each generation of the measurement system is a milestone in the lifecycle of the product (i.e., a project). The 1st generation of the measurement systems was an automated status reporting, which caused the number of measures and indicators to increase. The 2nd generation of measurement systems included a prioritization of indicators and wide spread of base measures. Adding mechanisms of gadgets (as previously described in Figure 3) caused even wider spread of measures. The 3rd generation of measurement systems was aligned with other software development programs, which reduced the number of measures but also reduced the information provided by the measurement systems. The major driver for the latest increase in the number of measures is the fact that software development teams started working with local, personalized measures, that is, measures which are designed and valid only for specific teams or disciplines, not for the company or for the software development projects.

The trend described above was common for all software development projects for the studied unit (although the actual numbers varied). Extrapolating this trend gives an overview of what the situation can look like for the entire unit. Given this trend the centralized measurement infrastructure could not scale, which was addressed by the MetricsCloud.

(2) *Interviews.* We conducted interviews with a number of experts working with measurement systems: (i) measurement program leader with over 7 years of experience in designing, implementing, and managing measurements at the company, (ii) measurement system designer with 3 years of experience in designing measurement systems, (iii) measurement system designer with 1 year of experience in designing measurement systems, and (iv) randomly selected stakeholder for the measurement systems with over 1 year of experience in setting requirements and using measurement systems.

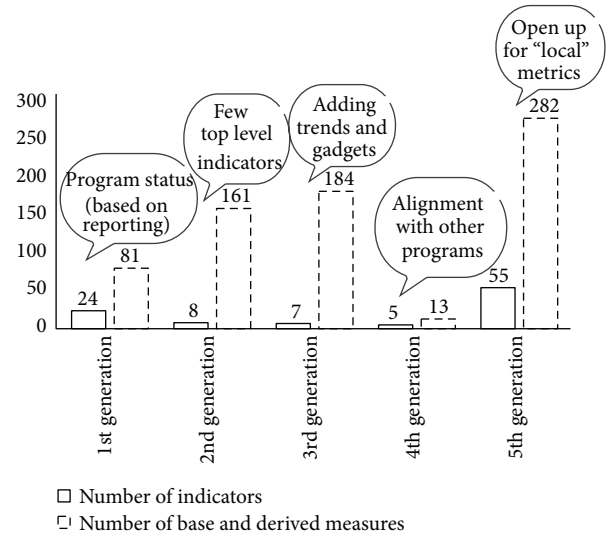


FIGURE 9: Growing number of indicators in one of the software development programs at Ericsson.

When discussing the positive and negative aspects of web-based dissemination of information the team has recognized the following issues:

(i) positive aspects:

- (a) open access to measurement information for all stakeholders in the organization: the public measurement systems were accessible to all employees; such openness stimulates benchmarking, comparison, and sharing good practices across the company;
- (b) centrally maintained execution environment: thanks to in-house developed execution software a small number of engineers of the metric team could execute and monitor large number of files of measurement systems;
- (c) common access to files: all measurement systems were stored centrally and accessed via web pages;
- (d) common naming conventions which simplified cataloguing of measurement systems;

(ii) negative aspects:

- (a) maintenance and development of measurement systems with limited resources in the metric team posed a limitation for the organization; the complexity of the infrastructure made it difficult for the stakeholders to easily place their own measurement systems in the common storage; a simpler solution would allow the stakeholders to manage their own measurement systems and “push” them to share with others as public;
- (b) rigid data flow made the infrastructure viable to failure propagation;

- (c) the main limitation of the current measurement infrastructure was perceived to be the dependency on MS Excel; almost all measurement systems were MS Excel files (the minority included Ruby/Python scripts or PHP pages).

The interviews after introducing MetricsCloud showed the following:

(i) positive aspects:

- (a) seamless updates: the measurement systems were synchronized seamlessly with simple notifications, which helped the stakeholders to focus on their business objectives rather than accessing the measurement systems;
- (b) the open access was kept and the stakeholders got a simple and immediate access to the measurement systems;
- (c) complementing the centrally managed measurement system with the scaled-down execution of private measurement systems on stakeholder's computers allows the stakeholder to take full control of the update of measurement systems (if needed);
- (d) common access to files: all measurement systems were stored centrally and accessed via web pages;
- (e) common naming conventions simplified cataloging of measurement systems;
- (f) the stakeholders were able to add their own measurement systems and share them with others;
- (g) the metric team could potentially reduce the effort of keeping the web-based infrastructure available all the time as the information was still available on stakeholder's computers (through the synchronization); short breaks in availability were no longer as problematic to the organization as before;

(ii) negative aspects:

- (a) new communication patterns: since the cloud-based dissemination was a new approach, the stakeholder interviewed was concerned about the security aspects of the disseminated information, for example, the need for solutions as in [19]; this was identified as the further work for MetricsCloud;
- (b) failure propagation: the new dissemination patterns require new models of handling of information quality of measurement systems. Dissemination of information to clients needs to be preceded with ensuring of quality in a more strict manner. Since the synchronization is controlled by distributed clients, the central infrastructure has to ensure that the information to be distributed has the right quality, for example, [11].

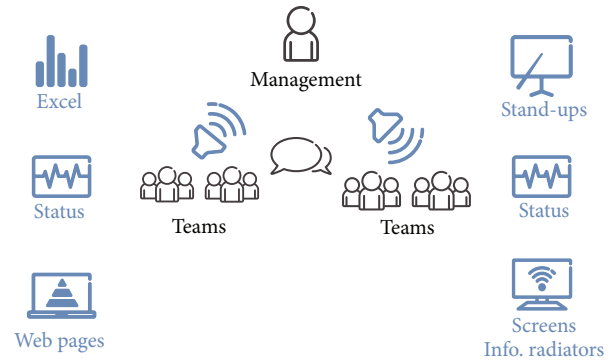


FIGURE 10: Channels for disseminating metrics in the organization.

Based on the evaluation we concluded that using cloud-based systems for dissemination of information separates the concerns of stakeholders (information) and the metric team (execution and infrastructure). This increases the involvement of stakeholders in creating their own measurement systems and spreading them in the organization using the common infrastructure. This type of dissemination reduced also the bottlenecks in creating and managing measurement systems by the metric team. The metric team could focus on more advanced measurement systems.

Based on the evaluation we have also defined types of measurement systems based on the metrics dissemination patterns presented in Figure 10.

(3) *Access Statistics*. One aspect of the evaluation of the importance of the cloud-based solution is the measurement of how often files are accessed. For this purpose we use the statistics of number of fetches of measurement systems which is collected by the measurement infrastructure at Ericsson. The statistics is checked for those measurement systems which contain indicators and are available publicly (i.e., not the private and not the shared measurement systems). We select a sample of 180 measurement systems which are representative for the entire set of measurement systems and which present different type of access patterns. The statistics were collected over a period of 28 weeks.

Figure 11 presents a histogram of access statistics for the sample of 180 measurement systems. The majority of measurement systems are accessed between 0 and 10 times in the period, but there are measurement systems which are accessed more often (including a measurement system accessed over 260 times during the period of 28 weeks). Due to confidentiality agreements with our industrial partner we changed the names of the measurement systems to “measurement system XX.”

The total number of fetches for these measurement systems shows only one side of the complexity; the frequency of access per week is the other side. Figure 12 shows the access statistics per measurement systems as a heatmap [32]. In the heatmap each row represents one measurement system and each column represents one week. The intensity of the color shows how many times the measurement system was accessed during the week. The figure shows an excerpt of 180

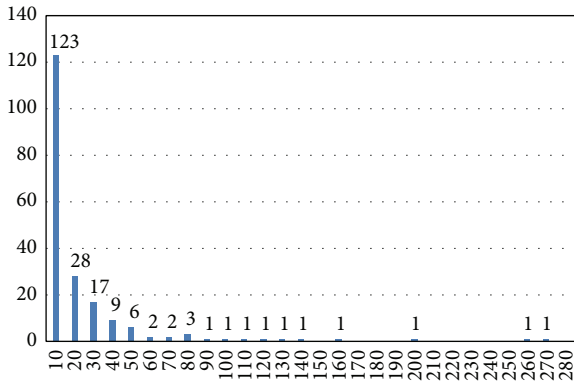


FIGURE 11: Access statistics for measurement systems.

measurement systems and contains measurement systems which are accessed regularly on a weekly basis (horizontal lines in the diagram, e.g., measurement system 4). This measurement system measures the status of the progress of feature implementation. It is updated daily and provides an insight into the development status of each feature of the product.

Another pattern is illustrated by measurement system 15. This measurement system measures the value flow in the organization and is accessed periodically as this piece of information is needed before specific periodic meetings at the company. Before these meetings the measurement system is used intensively and there are periods when it is not accessed (white spots in the line) but it is still updated.

Measurement system 1 is another example of usage pattern—using measurement system for a specific purpose for a limited time. This measurement system measures the test progress and defect status for one specific release. Since the release is developed during a defined period of time, this measurement system is of interest to stakeholders mostly during that time.

6.3. *Threats to Validity.* To evaluate the validity of our evaluation we follow the recommendations by Wohlin et al. [33].

The main *external validity* threat stems from the fact that we used a single organization to evaluate the approach. However, since the organization is mature in its measurement programs and use of measurement infrastructure, we believe that the results are generalizable to other mature organizations. The content of the feedback obtained showed such a maturity.

The main *internal validity* threat is related to the fact that we conducted an action research project [34]. In these kinds of projects the formulation of research questions can be prone to organizational context. In order to minimize the threat we have studied other types of systems based on the cloud technology and compared their requirements/functionality with MetricsCloud from the perspective of how general these are.

The main *conclusion validity* threat is the fact that we have used interviews as a means of collecting feedback on MetricsCloud as a dissemination technique. In order



FIGURE 12: Heatmap visualizing frequency of access of measurement systems per week.

to minimize the threat of respondent bias we used several interviews.

7. Recommendations for Other Companies

Based on the experiences from developing MetricsCloud and the experiences from previous work (e.g., introducing measurement systems using MS Sidebar gadgets [21, 35]), we can provide the following recommendations for other companies:

- (1) identify patterns of using metrics and the related types of measurement systems—design the measurement system based on the identified patterns and update strategies (e.g., local → public);
- (2) use both public and private measurement systems and separate them—provide the possibility for the stakeholders to share the measurement systems, but also to use some of the measurement systems for individual purposes;
- (3) provide the possibility to execute the measurement systems automatically—the users should be able to take advantage of automatization of measurement processes;
- (4) make the client as smart as possible—create the possibility to use attributes of file system, location of files, and other available pieces of information as context and make the client context-aware;
- (5) when introducing the cloud use analogies to existing systems, for example, Amazon Cloud or Dropbox;
- (6) use the approach of the minimum-viable-product [36]—this provides the possibility to evolve the use of metrics together with designing the cloud dissemination system.

Following the above recommendations increases the probability of a successful adoption of the dissemination system.

8. Conclusions

Measurement programs in large software development organizations are usually a blend of automatic and manual measurement systems. The majority of measurement systems are automated and are to be disseminated within the organization. However, some of the measurement systems are dedicated for specific stakeholders only or updated manually and disseminated to the entire organization. Development, management, and maintenance of a measurement program with such a diversity can quickly become costly and/or labor-intensive.

In this paper we describe how to address the scalability problem and how to overcome the main challenges of disseminating measurement information, for example, constant update of information, sharing of measurement systems, and separation of competence areas. We describe how cloud concepts of IaaS can be applied to support the variety of

uses of measurement systems in modern companies. We show which types of measurement systems exist, the related synchronization strategies, and the design of MetricsCloud—a system supporting these types and strategies.

By developing a cloud-based dissemination system—MetricsCloud—we reevaluated which dissemination patterns exist in the company and how to realize them. By introducing MetricsCloud into the organization we evaluated how the cloud-based dissemination can address the needs of the stakeholders and the metrics team. We organized the results into a set of recommendations for other companies that are willing to adopt similar solutions.

In our further work we intend to study how the use of MetricsCloud changed the patterns of using metrics at the studied company. We intend also to identify new patterns of suggesting metrics for stakeholders based on data mining of the storage of public metrics.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

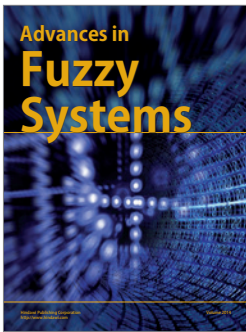
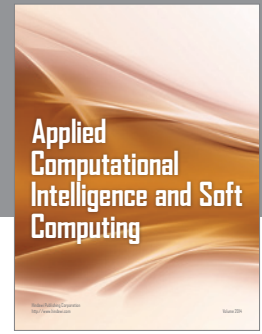
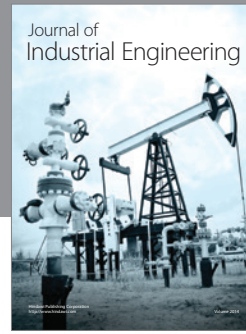
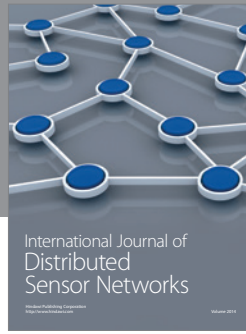
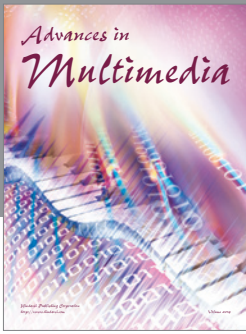
Acknowledgment

This work has been partially supported by the Swedish Strategic Research Foundation under the Grant no. SM13-0007 (Profiling Product and Organizational Performance).

References

- [1] O. Gassmann and M. Von Zedtwitz, “Trends and determinants of managing virtual R&D teams,” *R and D Management*, vol. 33, no. 3, pp. 243–262, 2003.
- [2] I. J. Chen, “Planning for ERP systems: analysis and future trend,” *Business Process Management Journal*, vol. 7, no. 5, pp. 374–386, 2001.
- [3] H. H. Olsson, H. Alahyari, and J. Bosch, “Climbing the “stairway to heaven”—a multiple-case study exploring barriers in the transition from agile development towards continuous deployment of software,” in *Proceedings of the 38th EUROMICRO Conference on Software Engineering and Advanced Applications (SEAA '12)*, pp. 392–399, IEEE, 2012.
- [4] M. Fowler and M. Foemmel, “Continuous integration,” ThoughtWorks, <http://www.thoughtworks.com/ContinuousIntegration.pdf>.
- [5] S. Chaudhuri, U. Dayal, and V. Narasayya, “An overview of business intelligence technology,” *Communications of the ACM*, vol. 54, no. 8, pp. 88–98, 2011.
- [6] H. Chen, R. H. Chiang, and V. C. Storey, “Business intelligence and analytics: from big data to big impact,” *MIS Quarterly*, vol. 36, no. 4, pp. 1165–1188, 2012.
- [7] S. Negash, “Business intelligence,” *The Communications of the Association for Information Systems*, vol. 13, no. 1, p. 54, 2004.
- [8] E. Whitworth and R. Biddle, “The social nature of agile teams,” in *Proceedings of the Agile Conference (AGILE '07)*, pp. 26–36, Washington, DC, USA, August 2007.
- [9] S. Bhardwaj, L. Jain, and S. Jain, “Cloud computing: a study of infrastructure as a service (IaaS),” *International Journal of*

- Engineering and Information Technology*, vol. 2, no. 1, pp. 60–63, 2010.
- [10] M. Staron, W. Meding, and C. Nilsson, “A framework for developing measurement systems and its industrial evaluation,” *Information and Software Technology*, vol. 51, no. 4, pp. 721–737, 2009.
- [11] M. Staron and W. Meding, “Ensuring reliability of information provided by measurement systems,” in *Software Process and Product Measurement*, pp. 1–16, Springer, New York, NY, USA, 2009.
- [12] P. Pawluk, B. Simmons, M. Smit, M. Litoiu, and S. Mankovski, “Introducing STRATOS: a cloud broker service,” in *Proceedings of the IEEE 5th International Conference on Cloud Computing (CLOUD '12)*, pp. 891–898, IEEE, Honolulu, Hawaii, USA, June 2012.
- [13] A. Shollo and K. Pandazo, “Improving presentations of software metrics indicators using visualization techniques,” Rapport nr.: Report/IT University of Göteborg 2008:020, 2008.
- [14] Y. B. Yoon, J. Oh, and B. G. Lee, “The establishment of security strategies for introducing cloud computing,” *KSII Transactions on Internet and Information Systems*, vol. 7, no. 4, pp. 860–877, 2013.
- [15] C. Vecchiola, X. Chu, and R. Buyya, “Aneka: a software platform for .NET-based cloud computing,” in *High Speed and Large Scale Scientific Computing*, pp. 267–295, IOS Press, 2009.
- [16] F. Liu, W. Guo, Z. Q. Zhao, and W. Chou, “SaaS integration for software cloud,” in *Proceedings of the 3rd IEEE International Conference on Cloud Computing (CLOUD '10)*, pp. 402–409, July 2010.
- [17] L.-J. Zhang and Q. Zhou, “CCOA: cloud Computing Open Architecture,” in *Proceedings of the IEEE International Conference on Web Services (ICWS '09)*, pp. 607–616, Los Angeles, Calif, USA, July 2009.
- [18] C. Gong, J. Liu, Q. Zhang, H. Chen, and Z. Gong, “The characteristics of cloud computing,” in *Proceedings of the 39th IEEE International Conference on Parallel Processing Workshops (ICPPW '10)*, pp. 275–279, September 2010.
- [19] S. Pearson, Y. Shen, and M. Mowbray, “A privacy manager for cloud computing,” in *Cloud Computing*, pp. 90–106, Springer, 2009.
- [20] P. Tomaszewski, P. Berander, and L.-O. Damm, “From traditional to streamline development—opportunities and challenges,” *Software Process: Improvement and Practice*, vol. 13, no. 2, pp. 195–212, 2008.
- [21] M. Staron, W. Meding, G. Karlsson, and C. Nilsson, “Developing measurement systems: an industrial case study,” *Journal of Software Maintenance and Evolution*, vol. 23, no. 2, pp. 89–107, 2011.
- [22] M. Staron and W. Meding, “Monitoring bottlenecks in agile and lean software development projects—a method and its industrial use,” in *Product-Focused Software Process Improvement*, pp. 3–16, Springer, 2011.
- [23] M. Staron, “Critical role of measures in decision processes: managerial and technical measures in the context of large software development organizations,” *Information and Software Technology*, vol. 54, no. 8, pp. 887–899, 2012.
- [24] A. E. Compute, *Amazon Web Services*, vol. 9, 2011.
- [25] G. Pallis, “Cloud computing: the new frontier of internet computing,” *IEEE Internet Computing*, vol. 14, no. 5, pp. 70–73, 2010.
- [26] B. Hayes, “Cloud computing,” *Communications of the ACM*, vol. 51, no. 7, 2008.
- [27] M. Armbrust, A. Fox, R. Griffith et al., “A view of cloud computing,” *Communications of the ACM*, vol. 53, no. 4, pp. 50–58, 2010.
- [28] I. Drago, M. Mellia, M. M. Munafò, A. Sperotto, R. Sadre, and A. Pras, “Inside dropbox: understanding personal cloud storage services,” in *Proceedings of the ACM Conference on Internet Measurement Conference (IMC '12)*, pp. 481–494, November 2012.
- [29] Z. Li, C. Wilson, Z. Jiang et al., “Efficient batched synchronization in dropbox-like cloud storage services,” in *Middleware 2013*, pp. 307–327, Springer, 2013.
- [30] A. Rosenthal, P. Mork, M. H. Li, J. Stanford, D. Koester, and P. Reynolds, “Cloud computing: a new business paradigm for biomedical information sharing,” *Journal of Biomedical Informatics*, vol. 43, no. 2, pp. 342–353, 2010.
- [31] M. Staron and W. Meding, “Using models to develop measurement systems: a method and its industrial use,” in *Software Process and Product Measurement*, pp. 212–226, Springer, 2009.
- [32] M. Staron, J. Hansson, R. Feldt et al., “Measuring and visualizing code stability—a case study at three companies,” in *Proceedings of the Joint Conference of the 23rd International Workshop on Software Measurement and the 8th International Conference on Software Process and Product Measurement (IWSM-MENSURA '13)*, pp. 191–200, Ankara, Turkey, October 2013.
- [33] C. Wohlin, P. Runeson, M. Höst, M. C. Ohlsson, B. Regnell, and A. Wesslén, *Experimentation in Software Engineering*, Springer, 2012.
- [34] P. Checkland and S. Holwell, “Action research: its nature and validity,” *Systemic Practice and Action Research*, vol. 11, no. 1, pp. 9–21, 1998.
- [35] M. Staron and W. Meding, “Transparent measures: cost-efficient measurement processes in SE,” in *Software Technology Transfer Workshop*, Kista, Sweden, 2011.
- [36] E. Ries, *The Lean Startup: How Today's Entrepreneurs Use Continuous Innovation to Create Radically Successful Businesses*, Random House, 2011.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

