A Metrics Model to Measure the Impact of an Agile Transformation in Large Software Development Organizations

Jeanette Heidenberg[1,2,3], Max Weijola[1,2], Kirsi Mikkonen[3], and Ivan Porres[1,2]

[1] Åbo Akademi University, Department of Information Technologies, Joukahaisenkatu 3-5 A, 20520 Turku, Finland
Givenname.Surname@abo.fi
[2] Turku Centre for Computer Science, Joukahaisenkatu 3-5 B, 20520 Turku, Finland
[3] Ericsson R&D Center Finland, Hirsalantie 11, 02420 Jorvas, Finland
Givenname.Surname@ericsson.com

Abstract. As the adoption of agile and lean methods continues to grow, measuring the effects of such a transformation can be valuable but challenging due to the many variables influencing the outcome of a software project. In this paper we present a metrics model developed for measuring the effects of an agile and lean transformation on software development organizations. The model was developed iteratively in cooperation with industry partners within the Cloud Software Finland research project. The resulting metrics model is applicable to projects of any size, complexity and scope, using metrics that support agile and lean values. The model can be used to measure both past and ongoing projects, regardless of whether the process model used is plan driven or agile. In order to evaluate the metrics model, the proposed model has been piloted in an industry setting.

Keywords: Metrics, Measurements, SPI, Transformation, Lean, Agile.

1 Introduction

Agile and lean software development methods keep growing in popularity among software companies of all sizes. Recent surveys performed by both academic researchers and IT consultants show agile and lean adoption ranging from 55%, as reported by Rodriguez et.al. [1], to 80%, as reported by VersionOne [2], in companies ranging in size from 11 to over 1000 employees [1].

The benefits of agile and lean deployment has been discussed extensively in the literature [3,4,5,6,7,8]; and so has the drawbacks of plan driven software development [9,6]. The discourse, however, mostly deals with the differences between the two ways of working in qualitative terms. The quantitative impact of agile and lean adoption in software organizations still needs further study [10].

Quantitatively and objectively comparing a development organization before and after an agile transformation is a challenging task. In this article, we use the term agile transformation to denote a sudden, disruptive change in the development process in an organization in order to adopt agile methods. In a large

H. Baumeister and B. Weber (Eds.): XP 2013, LNBIP 149, pp. 165–179, 2013.

scale agile transformation there are many factors affecting the outcome, and measuring the impact of only the transformation while excluding other factors is difficult. Since the transformation is performed in a real industry context, not in a controlled academic environment, there will inevitably be uncontrollable factors such as changes in the economic climate, shift in demand, and changes in the performance of the sales staff [11]. One can also expect uncontrollable factors internal to the software development organization, such as turnover of development staff, to impact the results.

Any transformation of the way of working may require a notable investment from the organization, both in a monetary aspect, as well as in disrupted working routines and possible resistance to change among some of the employees. Cohn [11] mentions phenomena such as waterfallacies and agile phobies as examples of such resistance. The organization may have embarked on the transformation with the goal of increasing the delivered value, and may after the transformation ask itself: *Was the transformation worth the effort?*. For this reason, an organization may look for a way of quantifying the impact of the transformation, despite the challenge of performing such measurements in an industry setting.

Indeed, this need for quantifying the impact of agile and lean transformations arose among the industry partners of the *Cloud Software Finland* research project [12] as a complement to already used qualitative measurements, such as interviews and surveys. It led us to formulate our research question: *"How can the changes of an agile transformation be measured by quantitative objective metrics?"*

In this article, we propose a software process improvement (SPI) metrics model for quantitatively comparing a software development organization before and after an agile and lean transformation. The model was iteratively developed and refined to focus on measuring efficiency, business value and lead-time with a number of industry partners within the *Cloud Software Finland* research project. The metrics proposed in the model are based on a *goal question metrics* (GQM) approach [13] where both the questions and metrics have been chosen and refined with a set of criteria to allow comparison between plan driven and agile. The questions and metrics were also chosen to continuously support the organization's agile transformation and new way of working.

The remainder of this paper is structured as follows. Section 2 gives a short background and we present previous work done in the field. In Section 3 our context and research method are described in more detail, including metrics selection criteria (Section 3.3). The goal for the transformation metrics model is presented in Section 4. Section 5 describes the partial literature survey undertaken to map the current state of agile and lean metrics. Our proposed metrics are described in Section 6. A pilot case and examples of the metrics in the model in an organization is presented in Section 7. The metrics are validated against the selected set of criteria in Section 8. Discussion and conclusion are presented in sections 9 and 10 respectively.

2 Background and Previous Work

Metrics in software development have been the subject of both research and practice for a long time resulting in many proposed best practices. Due to the introduction of agile and lean methods during the early 2000's, new best practices and research have been reported for metrics applied in agile settings [14,15,16,17,18,19].

Hartmann and Dymond suggest that agile metrics should affirm and reinforce lean and agile principles [14], since using inappropriate (plan driven) metrics can not only be inefficient but also threaten an emerging agile culture. Petersen and Wohlin discusses measurements used in lean manufacturing (e.g. Capacity Utilization), which are inappropriate in a software engineering setting due to the creative nature of the work [16]. Dubinsky et al. discusses the benefits of tailoring custom measurements for an organization as a means of communicating what behavior is considered important and therefore measured [15].

The topic of empirically comparing development processes has been identified as relevant by researchers. Mainly the reported findings have supported the deployment of agile methods. In the recent work of Concas et al. [20], software quality metrics were related with certain agile development phases and practices. However, the topic is still in need for further study [19,8], as very few reports on quantitative, empirical studies comparing the situation before and after an agile transformation exist.

One study that compares the effects of moving from a plan driven to an agile approach to software development is the case study by Petersen and Wohlin from 2010 [8]. The focus in the case study was on qualitative data gathered from interviews, with only a few quantitative performance metrics included for support. Our focus, however, is on quantitative data.

Another study comparing a transformation is the recent case study by Sjøberg, Johnsen and Solberg [19] focusing on the effects of a company migrating from Scrum to Kanban. They do use quantitative metrics, focusing on the variables lead time, quality, and productivity. The fact that they are mainly comparing two agile approaches limits the applicability of their research in our setting.

Two studies that we found useful for our work are Petersen's and Wohlin's work on flow [16], in combination with Staron and Meding's work on bottlenecks [18]. The main difference is in the purpose of the research. Their purpose was to continuously improve an agile way of working, by analysing the current way of working in order to find improvement opportunities. Our need for comparing the situation before and after a transformation is slightly different.

3 Research Question, Method and Context

3.1 Context

The need for a metrics model emanated from Finnish software companies collaborating with universities in the *Cloud Software Finland* [12] research project. One of the goals of this project is to support the Finnish software industry in

transforming their operations with the help of agile and lean methods. The size of the companies range from fairly large organizations, developing complex embedded software systems to smaller companies providing software development and consultancy services. Furthermore, both the software development processes and cultures of the companies in the project vary to a great extent. Some organizations have completed agile and lean transformations, whereas others are in different stages of transformation. Within this research project, there were already studies ongoing focusing on the qualitative measurement of agile and lean transformations, whereas our work pursues the quantitative aspect of the same transformations.

3.2 Research Question

As agile and lean methods are gaining popularity, we agree with previous work [10,19] that establishing a model for measuring the impact of a transformation is relevant for both researchers and practitioners. With this in mind and in the context described above, we formulate the following research question.

RQ - *How can the changes of an agile transformation be measured by quantitative objective metrics?*

As an answer to this question, we create a metrics model to be used for measuring an agile transformation. In order to ensure that the metrics model is useful for comparing the situation before and after a transformation, we set up the following criteria for selecting the metrics of the model.

- *C1.* The metrics must be applicable to both plan driven and agile projects.
- *C2.* The metrics must support the agile principles (as described in the agile manifesto [21]).
- *C3.* The metrics must be feasible to collect for both past and on going projects.
- *C4.* The metrics must be possible to collect and use in projects of any scope, size and complexity.
- *C5.* The metrics must be objective, i.e. metrics colletion should not require the judgment and interpretation of experts.

3.3 Research Method

The metrics model in this paper has been iteratively developed in a series of workshops with both industry experts and researchers. Much valuable knowledge have also been obtained through literature study and the software research community.

The metrics model in this paper was developed in five distinct steps. As the first step, in early 2011, the goal for the model was defined and iteratively refined with respect to goal coordinates according to the GQM method [13] as described in Section 4. The second step involved refinement of the goal and probing for

candidate metrics during formal and informal workshops (Section 4). When proposals for the goal and questions were available, as the third step a literature survey was undertaken to map the state of the art in agile and lean transformation metrics. The literature survey is described in Section 5, with the goal of discovering metrics already proven useful and relevant in literature. The fourth step consisted of iteratively evaluating the metrics discovered in the literature survey against both the questions in the first and second step, as well as against the selection criteria presented in the research question in Section 3.2. Finally, as the fifth step, when both researchers and industry representatives reached an agreement on the metrics to be included in the model, a pilot evaluation was performed to exemplify the data gathering and visualization of a subset of metrics at Ericsson R&D Center Finland, further described in Section 7.

4 Defining the Goal and Questions for the Transformation Metrics Model

In this section we describe the background to the GQM [13] approach as the foundation to the metrics model. We start with the background leading to the model, followed by an overview of the goal and questions derived from collaboration with industry partners.

4.1 Background and Development of the Model

Based on the needs and state of the industrial partners in the Cloud Software Finland research project a measurement model was jointly developed and iterated using a GQM [13] approach. Our main partner in the early iterations of the metrics model was Ericsson R&D Center Finland and as the model matured and was refined other partners joined the research effort.

In this work we have based our transformation metrics model on agile metrics best practices from literature to support the agile transformation during the measurements, as described in Section 2.

4.2 Goal

The general goal of transforming and improving development operations was identified in a project wide survey in 2011 (see Section 3.1). This goal was further discussed and defined in 2011 within the case company and subsequently iterated to clarify the goal coordinates *issue*, *object* and *viewpoint* as described by Basili et al. [13]. The beginning stage of defining the measurement goal consisted of a workshop day with open discussions.

From a later workshop, held in a world café [22] format, the GQM *coordinate issues* of the goal, **business value delivery** and **efficiency** were extracted to be of key importance. The workshop participants represented many different roles in the organization, such as: Scrum masters, developers, testers, product owners and line managers. The last coordinate issue, **end-to-end lead time**, was at

the time of the workshop counted as part of efficiency, but was later discovered to be more important and subsequently separated.

The final GQM goal coordinates are presented below:

Purpose: Improve
Issue: End-to-end lead time, business value delivery and efficiency
Object: The software development process
Viewpoint: From the whole organizations and customer viewpoint

The iterations and refinement of the goal coordinates were finalized during fall 2011 resulting in the following goal:

Goal: *Improve end-to-end lead time, business value delivery and efficiency for the software development process from the whole organization's and customer's viewpoint.*

When improving the lead time, business value delivery and efficiency, there is a risk that the quality suffers. In order to ensure that this has not been the case, we also added quality as an issue to measure.

4.3 Questions

Based on the goal and aforementioned coordinate issues, four questions were proposed:

Q. 1: *Are we more responsive in the new way of working?*
Q. 2: *Do we have better throughput in the new way of working?*
Q. 3: *Do we have a better workflow distribution in the new way of working?*
Q. 4: *Do we have better product quality in the new way of working?*

4.4 Metrics

The final step in the GQM modeling was to determine the metrics to be used. This step proved to be the most challenging and time consuming. According to Basili et. al. [13] one important factor in choosing metrics is to maximize the use of existing data sources. The use of existing data sources was particularly important in the research setting since the researchers were separate from the measured organization and the transformation was already taking place.

As input towards finding the right metrics for each question, the researchers conducted a literature survey as described in Section 5 and analyzed the results. From the state of the art in metrics, best practices were extracted. These acted as a base for selecting and crystallizing the right metrics in the GQM approach for the current research setting.

The four questions and the eight metrics chosen for inclusion in the metrics model are described in more detail in Section 6. In the next section, Section 5, we describe the literature survey undertaken to map the current state of research in SPI and agile metrics.

5 Literature Survey

A literature survey was performed to map the current status of Software Process Improvement (SPI) evaluation and agile metrics. The literature research was performed by the researchers in October 2011 as an on-line search in the following collections: SpringerLink [23] , IEEE Xplore [24] and ACM Digital Library [25].

The search strings used were: Lead-Time AND Lean, Lead Time AND Lean, Business Value AND Lean, Business-Value AND Lean, Metric AND Lean, Metrics AND Lean, Metrics AND Agile, Metric AND Agile. The search terms were entered in each database manually, which can lead to some inconsistencies due to human error. The titles of the search results were evaluated and the abstracts of relevant publications were further examined. The relevant publications were saved and subjectively ordered by precedence with numbers from 1-3 depending on assessment of the researchers and the number of times the publications appeared in the search results.

After the on-line search phase was finished the abstracts of the saved publications (70 in total) were evaluated a second time and relevant articles were chosen for reading. This method of both systematically and subjectively eliciting relevant articles depended on the research environment where the research questions were not completely finalized from the beginning.

Additionally some research was conducted in on-line journals such as Agile Journal [26] to further broaden the understanding of the topics and related terminology from practitioners and the agile community.

Approximately 20 articles were chosen as relevant for the future work at the end of the literature research. These were studied in detail and annotated, also interesting references were further examined resulting in a big matrix of publications, their goals and metrics. The matrix was discussed by the researchers and candidate metrics were selected for the metrics model.

In addition to the concrete metrics discovered in the literature survey, many proposed best practices for agile measurement have been presented and discussed in literature [14,15,27]. The best practices were a valuable input in the work of creating the metrics model, although not all practices could be satisfied.

Subsequent workshops with industry representatives strengthened the potential use of some metrics, whereas other metrics proved difficult to gather from, e.g., old projects and had to be abandoned for this work. The resulting metrics for each question is presented in Section 6.

6 Proposed Metrics

The metrics for the model were chosen with the selection criteria, described in Section 3.2, in mind. Besides the applicability to both agile and plan driven settings, the availability and objectivity of the data are of importance.

Two metrics were chosen for each of the questions to add redundancy in case some metric would prove to not be possible to collect. The relatively high number of metrics also represents the different organizations that have been

collaborating in this work. When applying the model (as described in Section 7), organizations are encouraged to consider how many questions to try to answer and which metrics are possible to collect to answer these questions.

The metrics are defined by a textual description, the measured *attribute* and the actual *metric*, as described by Meneely et al. [28].

6.1 Q. 1: Are We More Responsive in the New Way of Working?

With this question we want to investigate if the response time has improved. This question relates to the *better end-to-end lead time* in the specified goal.

Responsiveness is often regarded as a key factor in software development. As Reinertsen [29] argues, in areas where response time is important, this is the only metric that should be used for improving service. One such important area is support operations where solving bugs and problems quickly is of high value to stakeholders.

Similarly, during development of new features, fast lead-time is important for numerous competitive advantages such as fast feedback loops and reducing the risk of requirements becoming outdated (waste), both examples of where decisions might decay over time [30].

Metric 1 – Customer Service Request (CSR) Turnaround Time. The first metric measures the turnaround for customer service requests. The metric is calculated from a timestamp when the request first comes to the development organization and from a timestamp when the request is resolved.
Attribute: Time
Metric: CSR date solved - CSR date created.

Metric 2 – Cycle-Time per Feature. The second metric measures cycle-time for features selected for development. Quick cycle-time is essential for competitive advantages as noted by Petersen [17]. The metric is calculated from a timestamp when the feature is added to the backlog and timestamp when the feature is ready for delivery. Cycle-time is regarded as a part of the lead-time. This metric also supports metric 4 – *Business value / Work effort*, since shorter cycle-time makes more frequent releases easier.
Attribute: Time
Metric: Feature delivery ready date - Feature added to backlog date.

6.2 Q. 2: Do We Have Better Throughput in the New Way of Working?

Where the first question concerned timeliness, this question aims to investigate whether the total amount of value delivered is greater in the new way of working during similar time periods and projects. The benefits of increased throughput have been discussed widely, including Andersson [31].

Metric 3 – Functionality / Work Effort. With the second question's first metric we want to measure how much functionality (also denoted as product

size [27]) that can be delivered in relation to a certain work effort. The proposed metric is the ratio of test points, as described by Dubinsky et al. in [15,27] divided by total time spent on the development measured in person hours. This metric (similarly to metric 1) also supports metric 4, since more functionality can be split into more frequent releases.
Attribute: Throughput
Metric: Test points / Person hours

Metric 4 – Business Value / Work Effort. Business value is measured as more frequent major releases [11] in relation to the work effort (person hours).
Attribute: Throughput
Metric: Number of major releases in a year / Person hours

6.3 Q. 3: Do We Have a Better Workflow Distribution in the New Way of Working?

The third question, concerning *workflow distribution* characterizes the new iterative way of working, as this is one of the goals for an agile transformation. Measuring the workflow helps the organization identify that a change in the way of working has indeed taken place.

Metric 5 – Commit Pulse. Commit pulse measures how continuous integration is within sprints [15,27] by counting the number of check-ins daily. The check-in data can be visualized in a diagram with days on the x-axis and number of commits on the y-axis. The aim is to have an even check-in pulse throughout sprints without high spikes of commits at the end of sprints.
Attribute: Regularity
Metric: Number of days between commits

Metric 6 – Flow. Measuring the flow in an organization supports responsiveness as proposed by Petersen & Wohlin [16], connecting also this metric back to the first questions. Having a continuous smooth flow without bottlenecks allows the development organization to better respond quickly to customer requests.
Attribute: Flow
Metric: Flow diagrams

6.4 Q. 4: Do We Have Better Product Quality in the New Way of Working?

With the three previous questions concerning the improvement of the development process, the final question takes into account the quality aspect of the product developed. Improvements in other areas must not take place at the expense of product quality.

Metric 7 – Number of External Trouble Reports (TR). External trouble reports are defect reports submitted from external users. This metric measures the total number of external trouble reports during a certain time period in a release of software in the old way of working compared to total number of external trouble reports from a similar project and similar time period in the new way of working.
Attribute: Amount
Metric: Number of external TR's originating from a certain release

Metric 8 – Days Open, External Trouble Reports. The final metric measures the average days external trouble reports have been unsolved from creation until solved. This metric is related to *Question 1: Are we more responsive...*, but it also measures the quality of the product. If trouble reports consistently take longer to solve, then it is likely that the defects found are more complicated or that the code base is more difficult to maintain. Both of these are indications that the quality of the product has deteriorated.
Attribute: Time
Metric: TR date solved - TR date created

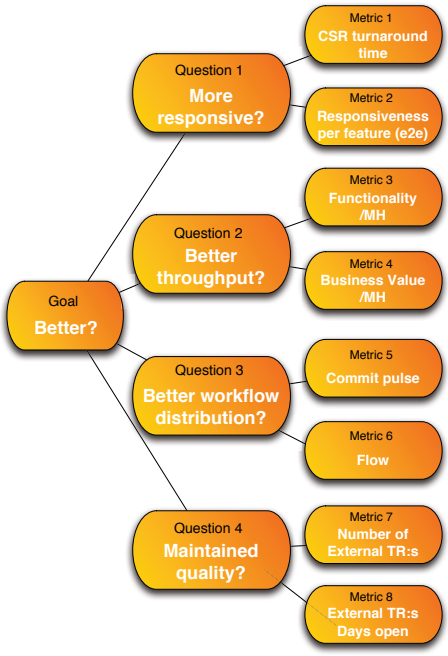
7 Using the Metrics Model in an Organization

To exemplify the use of the metrics model in an organization, data was gathered and analyzed from the agile and lean transformation Ericsson R&D Center Finland carried out during 2008-2011. Examples of these are displayed in Figure 1 and Figure 2, where Metric 1 and Metric 8 are shown respectively, with data plotted from both before, during and after the transformation. For the baseline, *old Way of Working (WoW)*, a set of features was selected from a typical development project in the plan driven development process from a two year period during 2007-2009. The *new WoW* was represented by a similar set of features from the agile and lean development process from a similar two year period in 2011-2012. Additionally some data was analyzed from the time (in 2010) when the transformation was taking place.

The collection of the data was considered cheap with respect to the amount of time necessary to extract the data by Ericsson R&D Center Finland representatives. The researchers gained access to the raw data files as well as were familiarized with the organization's terminology. Due to confidentiality, only trends can be shown in the metric visualizations, where all scales are linear and start the y-axis from zero.

8 Validation of the Model with Respect to the Criteria

In Section 3.2 we listed five criteria used for selecting the metrics of the proposed model. In this section we explain why we consider these criteria to be fulfilled by the included metrics.



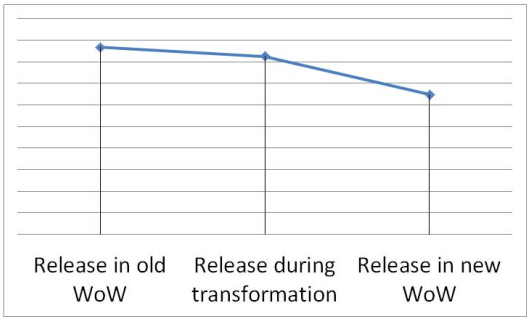


Fig. 1. Metric 1: Average number of days open for CRSs

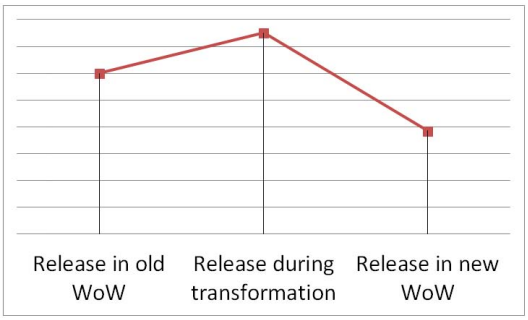


Fig. 2. Metric 8: TRs average days open

8.1 C1 and C4: Plan Driven and Agile Projects, Independent of Scope, Size and Complexity

With the exception of the workflow distribution metrics (Metrics 5 and 6 – commit pulse and flow), all the selected metrics are such that they measure the development effort from an external point of view. The time and effort for delivering service requests, features, functionality, business value, and trouble reports are measured looking at the point when they enter and exit the development organization. As such they ignore the internal process used to produce the result, and thus render the metrics independent of the internal process model used, i.e. plan driven or agile (C1). As the metrics do not consider the internal workings of the project measured, it is also agnostic to scope, size and complexity (C4).

In contrast, the workflow distribution metrics have the express purpose to illustrate the difference between the two process models, in order to verify that a change in way of working has indeed taken place. Metric 5 (commit pulse) is trivial to collect in any organization using a version control tool for the produced source code, regardless of the process model used. Metric 6 (flow) requires that the organization uses some form of time tracking tool for the development activities, and has continued to do so after the transformation. This may not always be the case.

8.2 C2: Support for Agile Values

As noted by Hartmann and Dymond, the inappropriate use of metrics can threaten an emerging agile and lean culture [14]. For this reason, we were

careful to select metrics that support agile values (C2). The core values and principles described by the agile manifesto [21] are centered around responsiveness, early delivery of working software, cooperation and communication, technical excellence, simplicity, self-organization and human interaction. Throughout the selection process, these values have served as a guide.

8.3 C3: Feasible to Collect for Both Past and Ongoing Projects

From a research point of view, it would be ideal to have the opportunity to define what metrics to collect a year before a transformation takes place. In reality, the need to measure a transformation arises at the start of the transformation at the earliest, and sometimes not until the transformation is complete. For this reason, the metrics defined should be feasible to collect after the fact. Any metrics initiative is a trade-off between cost of data collection and metric accuracy. In this case the trade-off becomes more pronounced due to the fact that it can be extremely costly to gather data for past projects. By working closely together with industrial partners and verifying that the data we ask for can be collected, we have achieved a model that is usable for both past and ongoing projects.

8.4 C5: Objective

All the data collected is quantitative and, with the exception of Metric 6 (flow,) requires no interpretation of experts. The entry and exit dates for implemented items as well as the number of items present should all be objective information present in the documentation of the projects measured. The flow metric is a slightly more complex metric that requires plotting and analysis of the plotted curve. This does not compromise the objectivity of the metric, but makes the implications of it slightly more cumbersome to analyze.

9 Discussion and Future Work

The metrics model proposed in this paper is intended to be a practical tool for use in software development organizations undergoing transformation from more traditional ways of working to agile and lean ones. The main challenge lies in the contrast between the old and new; in finding metrics that can be used in and accepted by both worlds. Especially the agile and lean philosophy rules out certain traditional productivity metrics such as lines of code per person hour or capacity utilization. This type of metric can, however, still be found in the agile literature. An example of a metric we chose not to use is churn (number of added, deleted and modified lines of code) per developer as presented by Sjøberg, Johnsen and Solberg [19]. We expected that this type of metric would meet with resistance and even be considered harmful in the agile way of working. Instead, we focused on measuring throughput in terms of the produced functionality and business value.

Another challenge in comparing the old and new way of working of an organization undergoing transformation is the fact that not all measurable changes originate from the transformation itself. In a real world situation, other factors will always impact the measurements. By choosing metrics that measure the operations of the development organizations (such as lead-times and defect reports), rather than metrics that measure the operations of the whole company (such as revenue and customer satisfaction) we hope to have minimized the effect of external factors. Internal factors, however, such as development staff turnover, cannot be factored out. This is a known limitation of this metrics model, which should be taken into account when the model is used. Any organization using the model should analyze and list the internal factors expected to impact the measurements. It is also worth noting that no metric model is immune to manipulation. For this reason, it is crucial that the collection of data be transparent, reproducable and honestly reported in order for the results to be trustworthy

The agile and lean community emphasises produced value as a measurement for productivity. The challenge we faced with this metric is that it is typically not collected in traditional ways of working. In fact, we found through our literature research that it is rarely collected in agile or lean ways of working. One could assume that the business value of a product could easily be extracted in hindsight by looking at the revenue produced for a product. This may work if the product catalog is simple, but proved to be impossible with the complex product portfolio of the software companies in the Cloud project. We have suggested a model for expressing business value for complex products in agile projects [32], but this proved costly to recreate after the fact for plan driven projects. The metric we used for measuring produced business value: the number of major releases, proved to be practical in both worlds, but can be argued to be a bit imprecise. This is an area we would still like to investigate further.

The metrics model developed here was created with the specific needs of our partners in mind. We are currently planning to investigate the general applicability of this model by running a series of case studies in different companies undergoing similar transformations.

10 Conclusions

In this paper we have proposed a metrics model for comparing development in plan driven processes to agile and lean development processes. The metrics model has been piloted with data from Ericsson R&D Center Finland, showing that the data needed for the model is indeed feasible to be collected both for ongoing, agile projects and past, plan driven projects and that the metrics are sensitive to an organizational transformation.

Our intention with the metrics model is twofold: 1) To contribute to current research in the field of comparing the effects of changing software development processes with a metrics model applied in an industry setting. 2) To support organizations by enabling them to show the benefits of agile and lean transformations with the use of quantitative objective data to complement qualitative studies in the field.

Our proposed metrics model consists of eight metrics, combined into pairs, connected to four questions, all aiming for one goal. Both the questions and the goal were iteratively developed during workshops in which both researchers and industry partners participated. The metrics were carefully selected to be compliant with a set of five criteria to maximize their usefulness in measuring agile and lean transformations.

Future work includes first and foremost applying the metrics model in different case companies to further validate the usefulness and applicability of the metrics model.

Acknowledgements. This work was funded by the CLOUD Software Finland project [12]. From Ericsson R&D Center Finland we would like to thank Christian Engblom, Kaisa Kettunen, Leena Pitkäranta, and Outi Väättänen for their help and support. We also wish to thank Marta Olszewska from Åbo Akademi for her participation in the project.

References

1. Rodríguez, P., Markkula, J., Oivo, M., Turula, K.: Survey on agile and lean usage in finnish software industry. In: Proceedings of the ACM-IEEE, ESEM 2012, pp. 139–148. ACM, New York (2012)
2. VersionOne: State of agile survey (2011), `http://versionone.com/pdf/2011_State_of_Agile_Development_Survey_Results.pdf`
3. Ebert, C., Abrahamsson, P., Oza, N.V.: Lean software development. IEEE Software 29(5), 22–25 (2012)
4. Šmite, D., Moe, N., Ågerfalk, P.: Agility Across Time and Space: Implementing Agile Methods in Global Software Projects. Springer (2010)
5. Anderson, D.: Agile Management for Software Engineering: Applying the Theory of Constraints for Business Results. The Coad Series. Prentice Hall (2004)
6. Leffingwell, D.: Scaling Software Agility: Best Practices for Large Enterprises. The Agile Software Development Series. Prentice Hall (2007)
7. Parnell-Klabo, E.: Introducing lean principles with agile practices at a fortune 500 company. In: Proceedings of the Conference on AGILE 2006, pp. 232–242. IEEE Computer Society, Washington, DC (2006)
8. Petersen, K., Wohlin, C.: The effect of moving from a plan-driven to an incremental software development approach with agile practices. Empirical Softw. Engg. 15(6), 654–693 (2010)
9. Royce, W.W.: Managing the development of large software systems: concepts and techniques. In: Proceedings of IEEE WESCON 26 (1970)
10. Dybå, T., Dingsøyr, T.: Empirical studies of agile software development: A systematic review. Inf. Softw. Technol. 50(9-10), 833–859 (2008)
11. Cohn, M.: Succeeding with Agile: Software Development Using Scrum. Addison-Wesley Professional (2009) ISBN 978-0321579362
12. Cloud Software Finland: Cloud software finland, `www.cloudsoftwareprogram.org`
13. Basili, V.R., Caldiera, G., Rombach, H.D.: The goal question metric approach. In: Encyclopedia of Software Engineering. Wiley (1994)

14. Hartmann, D., Dymond, R.: Appropriate Agile Measurement: Using Metrics and Diagnostics to Deliver Business Value. In: AGILE 2006 Conference (Agile 2006). IEEE Computer Society (2006)
15. Dubinsky, Y., Talby, D., Hazzan, O., Keren, A.: Agile Metrics at the Israeli Air Force. In: Proceedings of the Agile Development Conference, ADC 2005, pp. 12–19. IEEE Computer Society, Washington, DC (2005)
16. Petersen, K., Wohlin, C.: Measuring the flow in lean software development. Software: Practice and Experience 41(9), 975–996 (2011)
17. Petersen, K.: An Empirical Study of Lead-Times in Incremental and Agile Software Development. In: Münch, J., Yang, Y., Schäfer, W. (eds.) ICSP 2010. LNCS, vol. 6195, pp. 345–356. Springer, Heidelberg (2010)
18. Staron, M., Meding, W.: Monitoring bottlenecks in agile and lean software development projects – A method and its industrial use. In: Caivano, D., Oivo, M., Baldassarre, M.T., Visaggio, G. (eds.) PROFES 2011. LNCS, vol. 6759, pp. 3–16. Springer, Heidelberg (2011)
19. Sjøberg, D.I., Johnsen, A., Solberg, J.: Quantifying the effect of using kanban versus scrum: A case study. IEEE Software 29, 47–53 (2012)
20. Concas, G., Marchesi, M., Destefanis, G., Tonelli, R.: An empirical study of software metrics for assessing the phases of an agile project. International Journal of Software Engineering and Knowledge Engineering 22(04), 525–548 (2012)
21. Agile Alliance: Agile manifesto, `www.agilemanifesto.org/`
22. World Cafe: World cafe, `http://www.theworldcafe.com/method.html`
23. Springer: Springer link, `www.springerlink.com/`
24. IEEE: IEEEXplore, `www.ieeexplore.ieee.org/`
25. ACM Digital Library: ACM Digital Library, `www.dl.acm.org/`
26. Agile Journal: Agile journal, `www.agilejournal.com/`
27. Hazzan, O., Dubinsky, Y.: Agile software engineering. Undergraduate topics in computer science. Springer, Berlin (2008) ISBN: 978-1-84800-199-2
28. Meneely, A., Smith, B., Williams, L.: Validating software metrics: A spectrum of philosophies. ACM Transactions on Software Engineering and Methodology
29. Reinertsen, D.G.: The Principles of Product Development Flow: Second Generation Lean Product Development. Celeritas Publishing (2009)
30. Cockburn, A.: What engineering has in common with manufacturing and why it matters - ac (September 2006)
31. Andersson, D.: Agile management for software engineering: applying the theory of constraints for business results. Pearson Education Inc. (2004)
32. Heidenberg, J., Weijola, M., Mikkonen, K., Porres, I.: A model for business value in large-scale agile and lean software development. In: Winkler, D., O'Connor, R.V., Messnarz, R. (eds.) EuroSPI 2012. CCIS, vol. 301, pp. 49–60. Springer, Heidelberg (2012)