CHALMERS

Developing a roadmap of contextual factors and their impact on software measurement process efficiency - an industrial case study

Master of Science Thesis in Software Engineering and Technology

KRISTIAN MATTSSON

Chalmers University of Technology
University of Gothenburg
Department of Computer Science and Engineering
Göteborg, Sweden, 20/06/2011

Developing a roadmap of contextual factors and their impact on software measurement process efficiency - an industrial case study

Supervisor: Miroslaw Staron

Chalmers University of Technology
University of Gothenburg
Department of Computer Science and Engineering
SE-412 96 Göteborg
Sweden
Telephone + 46 (0)31-772 1000

Department of Computer Science and Engineering
Göteborg, Sweden June 3, 2011

Chalmers University of Technology
Kristian Mattsson mattssok@student.chalmers.se

Abstract

The utilization of efficient software measurement processes are highly valuable to an organization that strive towards producing high quality software. Nevertheless, an efficient software measurement process is a complex task, e.g. 80% of all software metrics initiatives fail and there are a number of industry-related problems with software metrics. This case study addresses two measurement processes within a large software producing organization, investigating how to make existing software measurement processes more efficient. This study presents a roadmap that illustrates the contextual situation, i.e. the surrounding push and pull factors, and shines a light on coordination activities which would allow for a more efficient data collection. Also, three key factors; support, definition and refinement are elicited and elaborated, with the objective of identifying important areas for more efficient and long lasting software measurement processes.

Keywords: Software metrics, measurement, software process, roadmap, contextual factors, push and pull.

Table of contents

Chalmers University of Technology
Kristian Mattsson mattssok@student.chalmers.se

List of abbreviations

Abbreviation	Explanation
CMMI	Capability Maturity Model Integrated
CSI	Customer Satisfaction Index
CSV	Comma Separated Values
FP	Functional Points
GQM	Goal Question Metric
KLOC	Kilo Lines of Code
LOC	Lines of Code
MAM	Metrics Acceptance Model
QSM	Quantitative Software Management
RSM	Resource Standard Metrics
SEI	Software Engineering Institute

Chalmers University of Technology
Kristian Mattsson mattssok@student.chalmers.se

1 Introduction

It has always been hard for companies to monitor and control the quality of their developed software. Developing software is a complex task that often is done by complex individuals that strive towards a continuously changing goal due to changed user requirements [1]. One result from this complex task is inadequate software quality due to inadequate development practices, something which costs the US industry about $60 billion per year [2]. In addition, according to a research report by Dynamic Markets [3], 62% of all software projects overran their estimated development time and 49% suffered from budget overruns. Also, over the life cycle of a typical software, about 50% of the total cost is attached to finding and repairing defects [4]. From these facts it can be argued that by improving software processes, ergo software quality, there are substantial benefits to be made. Furthermore, studies show that improvement activities lead to enhanced software quality and an overall better process [5]. Also, one highly important aspect, when determining the outcome of software process improvement activities, is the presence of a software metric program [2].

Software metric(s) is a term used to describe a wide number of activities that are focused towards quantifying software engineering, i.e. activities that are meant to measure the outcome and progress of a software product, process or project. The activities can vary between generating numbers from the software development, to producing models that assist when predicting software resource requirements and quality [6]. Tom DeMarco said, in [7], *"You can't control what you can't measure"* and that is the main reason behind software metrics, to be able to quantify and control software and its surrounding context.

Every company that strives for higher software quality and process improvement have a software metric program in place, and companies without metric programs usually produce software of a marginal level at best [4]. However, there is a noticeable difference between having a software metrics program and effectively making use of that program. Industry experience has revealed a number of problems concerning software metrics [4] and Rubin [8] points out that 80% of all software metrics initiatives fail.

Due to the fact that quality issues still pose high costs for the software industry, and that a majority of the started software metric initiatives fail, this study aim to investigate:

1. How two software measurement processes within an organization that already works with software metrics are affected by contextual factors?
2. How do internal pull factors contribute to assuring efficient data collection in the long run?

The questions are meant to explain: (1) the relationship between a measurement processes and its surroundings, i.e. how do the contextual factors affect the usage, quality and relevance of a software measurement process. (2) How stakeholders

within an organization should act to ensure maximum value and longevity of the measurement processes.

The studied organization, Amadeus, is a large software intensive company working in the travel industry. This case study is based on work done during a five month internship at an Amadeus site in Nice, France. The internship provided good insight into the organization and the divisions that work with software metrics.

The methodology used, to be able to answer questions (1) and (2), was to collect data from three different sources with the purpose of triangulate the data: First, semi-structured interviews, qualitative data collection, with team-leaders for the studied processes. Second, questionnaires, quantitative data collection, to enumerate how the workers responsible for the respectively measurement processes perceived the situation. Third, reviews of internal artifacts to compare with the answers from the qualitative and quantitative data. The results were compiled using tabulation, looking for trends regarding internal push and pull factors. (More about the methodology and the case study design can be found in section 4.) The data collection is designed to be able to answer research question (1) with a roadmap that describes how the surrounding context, from a market need and technology push perspective, affect the two different measurement processes. Additionally, address research question (2) by deriving three general guidelines, with additional recommendations, for ensuring longevity and maximum value from a software measurement process.

This paper is structured as follows. The following section presents related studies and previous reports carried out within this field. Section three presents an overview of the studied organization. Section four presents the case study, how the study was designed and carried out. Section five presents the results from the study along with an extended analysis. Section six presents the validity evaluation of the study followed by section seven which presents the conclusions.

2 Earlier studies

The literature that is apposite to this work is case studies of software measurement programs, with the addition of behavioral and organizational factors. This section presents a brief overview of terms related to software metrics, measures and measurement programs as well as literature that has been important to this study.

2.1 Overview of software measurement and metrics

A software measurement is a quantified attribute from a software program, product or process. The measurement is the raw data that is related to a variety of elements from the software process. Software metrics, or indicators, are derived from software measures, they are quantifiable and used to compare the current state with past performance, estimates or to make future predictions. Metrics can also be used to collect data for identifying trends in the development environment, detect anomalies and to highlight points for improvement [9].

Software metrics can be collected in various ways since metrics essentially is quantifiable factors surrounding the software development. However, when establishing what to collect it is important to thoroughly consider the validity and the use of the collected metrics. Authors like Westfall [10] and Staron [11] stress the importance of collecting "useful" metrics, and their research and findings were used as a baseline, to compare with the findings from this case study.

2.2 Software metric programs, industrial case studies and success factors

The roadmap and guidelines suggested in this paper is meant to aid organizations that already have a measurement program in place. Hence, the following publications were investigated to elicit important factors surrounding organizational software measurement programs in general, and to not be constrained by Amadeus context.

- [12]: studied factors that are necessary for long term success of software metric programs. The case study highlights the need of constant change in the software metric program to adapt with the ever changing software projects. From the case study they identified three key elements for a successful metric program, the use of industrial standards, a significant experience base and research activities. Their experiences were used, in this study, when identifying important factors for a successful program and for the construction of the questionnaire.
- [13]: investigates the determinants of success of software metric programs. They measured success using two variables, use of metrics information in decision making and improved organizational performance. From over 200 data points they concluded the importance for software managers to start by focusing on the technical factors and provide incentives for the developers to use software metrics. This report is interesting when assessing the situation and way of work at the studied organization.
- [14]: develops a model to investigate the likelihood of a software metric program being accepted in the current organization. The model they developed is called Metrics Acceptance Model (MAM) and connects four important factors for metrics acceptance: ease of use, usefulness, control and attitude. Each of the four variables is positively correlated with intention. This model, or areas, can help organizations with their metrics process to include the significant stakeholders. The report identifies relevant areas when it comes to evaluate an organizational structure.
- [15]: focus on the unexpected difficulties that arise when implementing a software metric program, with the goal of collecting basic and straight-forward, software metrics. They underline the high cost and training that is needed to collect high quality metrics, the importance of communication over a wide range of organizational units and question the rationale behind having a metric program at all due to the effort needed to streamline it with the organization. All important factors to consider when evaluating the studied organization and provides a rationale for some of the decisions

around the software measurement processes within the studied organization.

- [16]: uses the Goal Question Metric (GQM) approach to design a company-wide metric program for an Italian software company. The most interesting finding is: the effect that the development environment has on the developer's productivity and that the only way to get consistency in the data collection is if the related activities obey a predefined company-wide procedure. This report was used to consider the outcome from a similar study and match similarities with their experiences.
- [11]: designs a framework for software metric collection in a real industrial context based on the ISO 15939 standard. An interesting finding was that the framework, through successful implementation, managed to change the company culture and their view of software metrics. Also the importance of defined processes and roles around the collected data, to be able to present unbiased results. Further, the importance of an automated collection process is identified. Their experiences have been used through this report when discussing important factors for software metrics.
- [17]: investigates and links the internal success factors in measurement programs with the external success factors that exist in a larger organizational context. The external success factors are critical for creating real value for the organization and are key to solve the problems that arise in organizations, when implementing software metric programs. The report provides further insight into a large organizational context and which factors that affects the work with software metrics.

The above mentioned literature provides good general knowledge about factors directly, and indirectly, related to organizations and software measurement programs. The following section, 3, aims to offer deeper insight to the studied organization and the selected objects for this study.

3 Organizational Context

This section provides a brief overview of the current organizational structure at Amadeus, hereafter referred to as 'the organization'. The software development within the organization stretches from maintenance of existing products and systems to development of new products of varying size and complexity.

The objects for this case study are two software measurement processes that resides under two different organizational contexts. One process is maintained by a group that primarily work for top management and whose main focus is to provide software metrics for the whole organization, the organizational group is hereafter referred to as 'Group A'. The other process is mostly aimed to support developers and other divisional stakeholders; the process is maintained by a group that work as a service team towards one of the largest development divisions in the organization, the divisional group is hereafter referred to as 'Group B'. The process maintained by Group B is centered on collecting meta-data for the division which is used to assure a high quality development environment, for the developers. An organizational map, figure 1, illustrates how Group A and Group B are situated in the organizational

context. Since the context for these groups differs, a further explanation of their respectively situation follows.

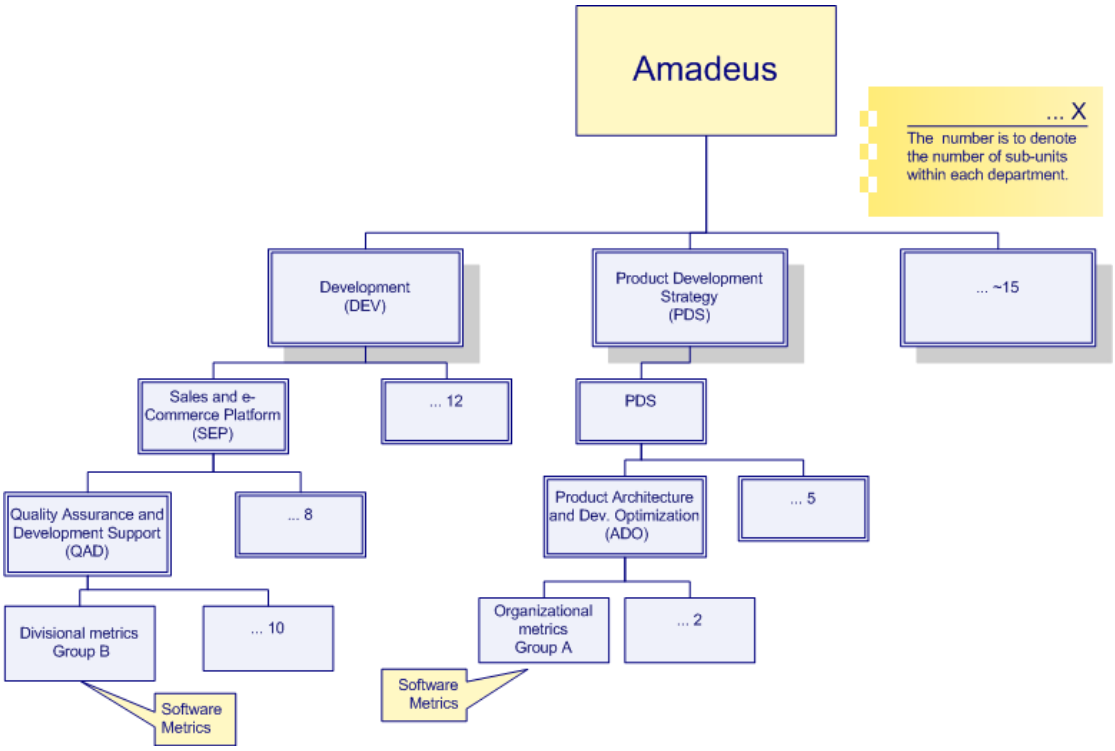

Figure 1. Organizational chart which describes where in the organization the two samples are situated.

3.1 Group A

The main responsibilities of Group A are:

- To distribute KPI's (Key Performance Indicators) on software development metrics.
- To increase efficiency and collaboration between the development divisions.
- To spread software development best practices.
- To maintain a knowledge-base on the tools related to the product-cycle.

However, currently Group A is mainly devoted to collecting and analyzing software metrics (KPI's), which is the area of focus in this section and study.

Group A became a dedicated software metrics group in the end of 2009; it was due to a request from top management about figures (metrics) about the organizations code repository. Though, software metrics was not something completely new at the time, the predecessor of Group A worked with the team responsible for system planning and did provide top management with metrics. But, the metrics was ad-hoc which presented problems, e.g. the analyzed results presented a big variance due to high reliance on the divisional input. Also, there was no formal data collecting process in place.

Group A began their current data-collecting process in 2010 with their own dedicated database for unbiased data storage of the analyzed metrics. Since the first installment of their current data collecting process, Group As activities centers on refining the collecting process and to answer requests from top management. Group A collaborates with top management and other key stakeholders since their main objective is to provide them with decision support. In addition, Group A aim to provide a macro level view of the organizational code status where high-level trends can be identified for deeper analysis. Thus, Group A is a reactive unit that works dynamically with the feedback they get from top management.

3.2 Group B

Group B is a service team and has a wide area of responsibilities. However, their most vital task is to provide support to the developers within the division. The support is mainly centered on refinement of the development environment, the maintenance of development tools and to establish procedures around the development of their current products. Hence, Group B is not solely dedicated to software metrics. Nonetheless, in their efforts to enhance and improve the development environment, which includes further support of the developers, they started to collect statistics regarding the developer environment and the code quality. They established a process to collect and analyze statistics regarding the code quality; also, they have custom made tools that collect data about the development environment. A deeper explanation about their collecting process can be seen in section 5.2.

Due to the fact that Group B serves as a service group, within a division, the collected data is at a micro detail for a low-level divisional overview. Group B works in an isolated setting, mainly with divisional stakeholders, and the majority of their activities are centered on the individual developer. The data collection processes of Group B are financed from divisional priorities, i.e. the current need for statistics regarding the development environment. Group B does not have any formal processes in place for spreading the environmental data they collect. The collected data that concern the code are visible for all stakeholders via an internal dashboard solution, but the environmental data is neither spread nor accessible outside the group. The reason for the data not being organizationally communicated is that it only concerns the context from which it was collected, and that there are no priorities from the division to spread the data.

4 Case study design

This exploratory case study investigates two of the software measurement programs that exists inside Amadeus with the purpose of pinpoint contextual push and pull factors that affect the development of software measurement programs. Software measurements are an increasingly important step towards high quality software development. Metrics are also part of industry standards such as ISO 9000 and the Software Engineering Institute (SEI) Capability Maturity Model Integrated (CMMI) [10]. In addition, industry standards such as ISO/IEC 15939 target how to conduct software measurements, furthermore, the Goal Question Metric (GQM) approach has become a standard for the definition of measurement frameworks [18]. The

GQM approach was originally developed by Basili and Weiss [19] and is designed to be a model to better define and interpret operational and measurable software. However, even though there are models and standards that exist around the area of software engineering, there are still different definitions of terms such as software metrics. Still, the studied software measurement programs, and their measurement processes, are used to ensure the (i) overall code quality, (ii) support process and (iii) product improvement. More details about the measurement programs and their group wise contexts are presented in section 5.1 and 5.2.

In this case study, roadmapping is used as a theoretical framework for describing the factors affecting the measurement program with strong focus towards the concepts of market pull and technology push.

4.1 Research questions

This exploratory case study intends to address the following research questions:

- How two software measurement processes within an organization that already works with software metrics are affected by contextual factors?

The question is important in order to explore what kind of factors pull the development of the measurement program. Since, according to an established roadmapping theory [20], the pull factors usually come from the users/market, the "market" and "user" is referred to as the context. The above question is answered by a roadmap describing the relationship between the market need and the technology push initiatives in the studied organization.

Establishing a measurement program is only a part of the success from an industrial perspective, executing and evolving it over a longer period of time is another part. Therefore the following research question is addressed:

- How do internal pull factors contribute to assuring efficient data collection in the long run?

In this context the "long run" is considered to be a period that stretches over the initial adoption phase and where the program is continuously refined and used to create value for the organization. Those factors are identified as the long run due to the fact that a majority of software measurement programs falter after the initial adoption phase [14] and for a measurement program to be successful it needs to create added value for the organization [17]. This question is answered by three factors elicited from the measurement processes in the studied organization.

4.2 Objects

The objects in this case study are two independent software measurement processes that are maintained by two different groups within the organization. Both measurement processes have been in place for about 6-12 months and are continuously enhanced and refined and are differentiated due to their diverse organizational contexts.

One of the processes serve to provide metrics from the whole organization and is maintained by a group, hereafter referred to as Group A, who work as a reactive software metric unit whose goal is to collect high level software metrics and provide top management with decision support. Group As process has been designed to collect unbiased data on a high organizational level which helps managers to get an overall picture of the current organizational situation and raise overall awareness. In contrast, the other studied process serves in a divisional context and is maintained by a group, hereafter referred to as Group B, which function as a service team in one of the biggest development divisions in the organization. The objective with Group B and their measurement process is to ensure a high quality development environment and to assist the developers within. This requires providing divisional stakeholder with detailed information and identifying possible degradation in the development environment.

Since the two processes are used, and designed, for different contextual environments serving different objectives, the interesting factors are the similarities and discrepancies between them.

4.3 Sample

The qualitative data for this analysis was chosen using convenience sampling, which focused on interviewing people in the groups, Group A and Group B, with deep knowledge and involvement in their current software metrics process. The team leaders of each group were interviewed:

- Team-leader for Group A works with the organizational metric program. The interviewee has long-term experience in measurement in the organization and has worked with this studied process since the start (end of 2009).

- Team-leader for Group B works with the divisional metric program that oversees all the activities carried out by Group B. The manager has experience from software measurement in the division and a wide range of additional activities due to the role of Group B. The manager is mainly focused towards development environment improvement activities which constitutes work with their measurement process.

These roles cover the adequate knowledge basis to collect the main source of the qualitative data for this study since both have years of experience regarding their own situation and context. Also, from their leading position they possess a good overall picture of their functions and limitations, and due to the fact that this study aims to describe contextual factors that can affect or hinder an already established measurement process the sample selection is highly capable of providing appropriate answers.

4.4 Data collection procedures

The qualitative data for this study was collected in the form of semi-structured interviews. The interviews were recorded, with the interviewees consent, and transcribed to ensure the quality of the data. (The transcribed interviews can be found in the Appendix.) The interviews were designed to cover different aspects of their work and the transcribed versions were codified for a better overview of the subjects. The subjects were chosen from the key terms from the ISO 9000 standard (process and product) and further extended by adding subjects that indirectly relates to the existing codes. These results were compared using a tabulation format for easier analysis [21] and can be seen to their full extent in the Appendix, table 12. In addition, reviews of internal artifacts were done to gain better insight of their current way of working and monitor their process conformance [22].

To further support the qualitative results, a questionnaire in the form of an online survey was sent out to members of the two groups to collect data from a broader sampling. The questionnaire consisted of a list of questions with possible answers that ranged from 0-3, including Not Applicable (N/A), with the purpose of identifying how true the questions where (ranged from '*No*' to *'Yes, completely'*). The framework and questions were loosely based on a framework first developed by Jeffery and Berry [23] and further developed by Staron and Meding [12]. One example question can be seen in figure 2 and the complete list of questions in table 1.

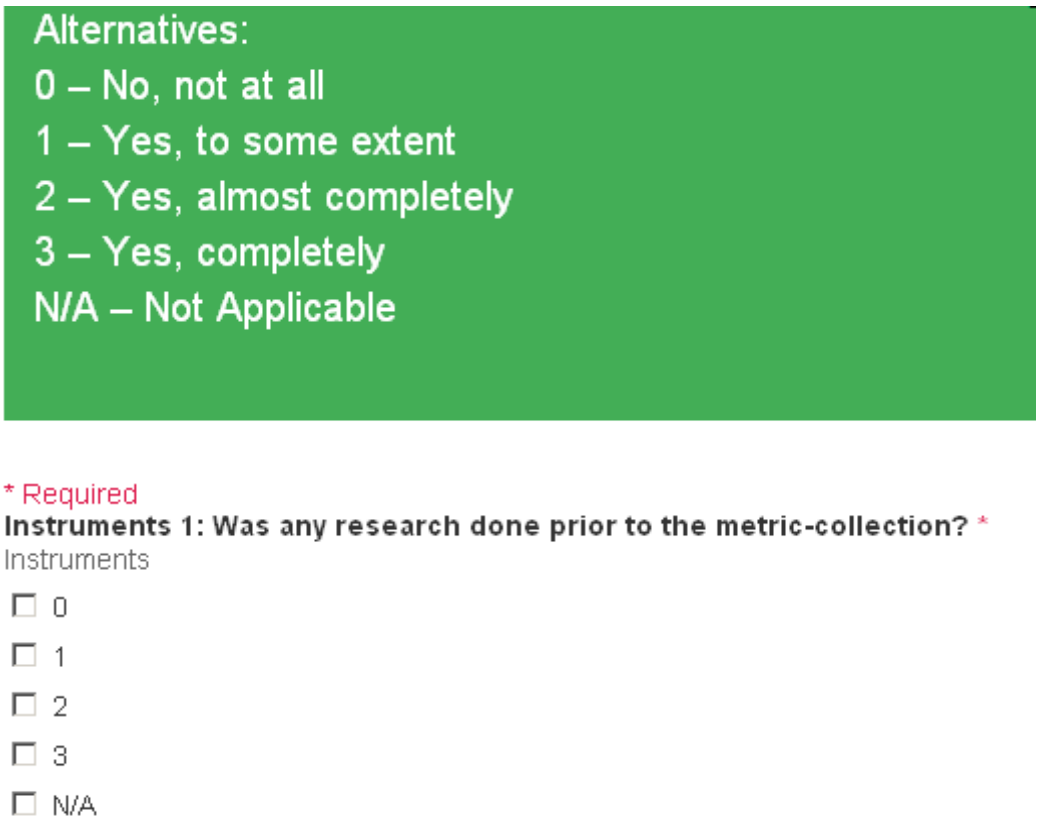

Figure 2. Example question to illustrate the structure of the questionnaire for the quantitative data collection.

	Instruments (I), Process (Pro), Product (P), Context (C)

1	I1: Was any research done prior to the metric-collection?
2	I2: Are ISO/IEEE standards used in the development/refinement of the metric collection?
3	I3: Is there training available in software measurement?
4	Pro1: Do the software metric-collection-process have sufficient resources?
5	Pro2: Is the goal of the software metric-collection-process clearly defined?
6	Pro3: Are top-management involved in the process?
7	Pro4: Are tools seen as a key factor in the software metric-collection-process?
8	Pro4.1: If so (Pro4), do you have sufficient resources for acquiring those tools?
9	Pro5: Are the sources for the different metrics trustworthy; i.e. the validity of the data behind the metrics?
10	P1: Is the outcome of the metric-collection clear? (Which metrics that should be produced and how they will be used)
11	P2: Is the results from the software metric-collection used by top-management?
12	P3: Are the results from the metric collection "pulled" by mangers? (I.e. is management interested in the collected metrics?)
13	P4: Do the current metric-product have enough respect from the organization (i.e. are the metrics used as decision support or are they just collected for the sake of collecting)?
14	P5: Are the collected metrics used to its full extent, i.e. are all the collected metrics used as support for some decision(s)?
15	C1: Are the goals with the measurements related to the business goals?
16	C2: Are there sufficient resources allocated for achieving those (the measurement) goals?
17	C3: Is the outcome of the data-collection clearly defined?
18	C4: Is there a planned pay-back period for the software-metric process (i.e. the metric-effort will give a good ROI in x years)?
19	C5: Is it clearly communicated in the organization/department what the software metrics is used for?
20	C6: Do the metric-process have the required support from top-management?

Table 1. Questions that were used in the questionnaire sent out for the quantitative data collection.

4.5 Analysis Procedure

The qualitative interviews were recorded and transcribed, the results from the interviews where fit into a partly pre-coded table with the codes 'process' and 'product'. The transcripts were reviewed for trends regarding contextual factors, such as push and pull, the formatted tabulation chart can be found in Appendix, table 12. The 'process' and 'product' sections were extracted and presented with related findings from internal artifacts to form a comprehensive baseline. In addition, the derived push and pull factors are highlighted through the baseline to display them in their context.

The qualitative data were analyzed by the use of descriptive statistics. Percentages with the total level of question conformance (max score 100%) is presented by a cross tabulation in table 9. The table-values were calculated based on the answer factor, e.g. the factor that represents total conformance is 3, on a sample size of 3 the total value that would represent a 100% conformance is 9. In addition, to test the overall variance between the two samples, a variant of the Customer Satisfaction Index (CSI) was applied. The CSI were calculated based on the number of answers in the top half (2-3) of the questionnaire, i.e. the total percentage of answers that was placed in the region of 2-3. For example, if all the respondents answered 2 the CSI

would be 100%, contrastingly, if half of the respondents answered 1 and the other half answered 3 the CSI would be 50%.

The derived push and pull factors were evaluated against the additional data from the descriptive tables and scientific literature and used to elicit three key factors for a successful data collection from a similar context.

5 Results and analysis

This section presents the results from this case study followed by an analysis. The section is structured as follows (i) results from the qualitative interviews and internal documents regarding 'Process', with subsections for the two objects. (ii) Results from the qualitative interviews and internal artifacts regarding 'Product', with subsections for the two objects. (iii) Statistically derived results from the quantitative interviews followed by roadmap supported by three key factors that address the two research questions.

5.1 Process

This section presents the results from the qualitative interviews of, Group A and Group B, and the internal artifacts about the software measurement process present in both units. In each section the identified push and pull factors are highlighted and compiled into a table, followed by a roadmap at the end.

5.1.1 Group A

There are two parts of the process that define how Group A works with software metrics. One technical part that constitutes how the group collects data, and one part that describes how Group A work with the different stakeholders to refine the technical process and its outcome. The current technical process can be seen below in figure 3.

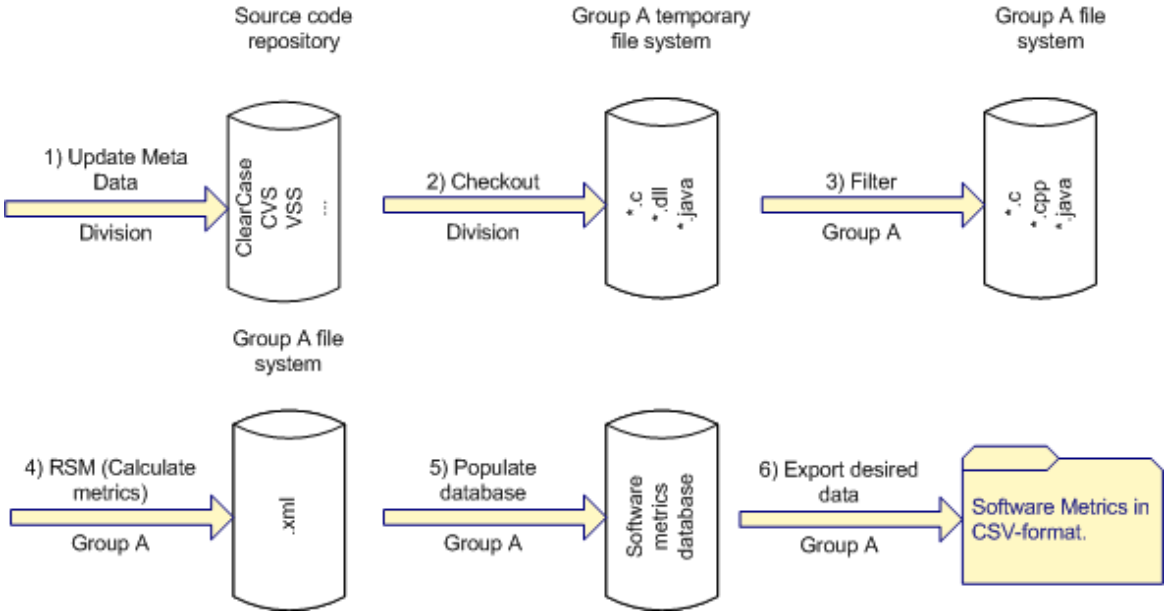

Figure 3. Overview of the data-collecting process used by Group A.

The technical process works as follows: (1) A development division update their source code repository with meta-data input. (2) The division triggers a checkout of the code to a temporary file-system, which is located in the domain of Group A. (3) A software called 'rsync[1]' is triggered by Group A to separate the files on programming language and store the outcome in a permanent file system. (4) A tool called Resource Standard Metrics (RSM) is applied to calculate software metrics, the output from RSM is stored as xml files that is used to populate (5) the database containing all software metrics. Group A are using stored procedures (6) to export that data as Comma-separated Values (CSV) files. The CSV files are read by a Microsoft Excel application and used for presentation; figure 4 illustrates two graphs from the presentation report.

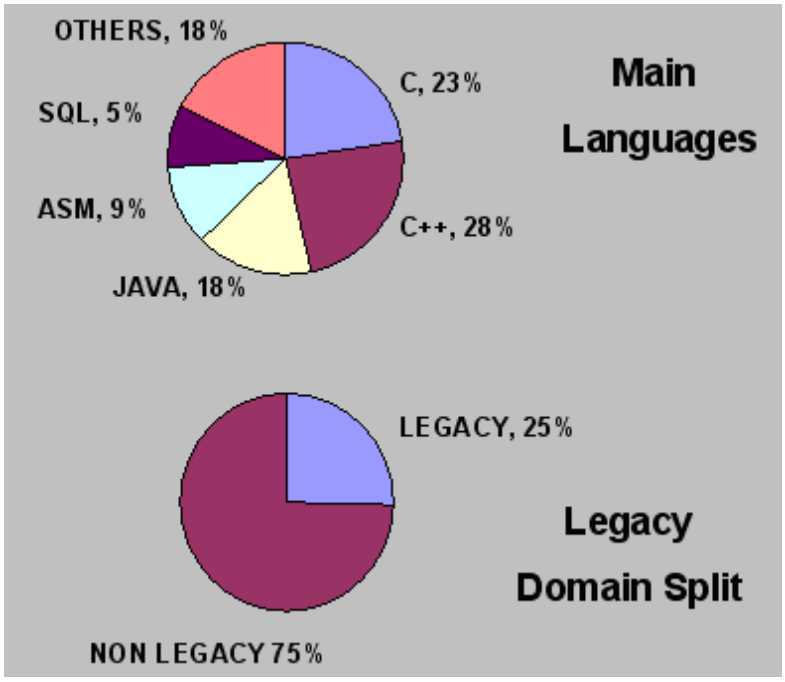

Figure 4. Example graphs from the presentation of measurements which describes the current language segmentation in the organization.

All the steps in the technical process need to be triggered by either a division-member or a representant from Group A (see figure 3). The complete process, from input of meta-data to software metric output, is technically executed in 2-4 hours. However, the time varies from 2 to about 48 hours depending on the quality of the input. The first step (1) is crucial since without good meta-data the outcome will not be valid. Hence, depending on the division that provides the meta-data, members of Group A need to go back and ensure the quality of the input. The step of needing to go back and manually validate the meta-data is, by far, the biggest bottleneck towards a fully automated data collecting process. The reason behind the big variance in execution time and the manual validation, depending on the division, is the internal systems divisions use for managing their code repository. Certain divisions have big amounts of legacy code and a big project portfolio which is stored

[1] rsync is a software application which provides incremental file transfer; it synchronizes files and directories from one location to another. See rsync.samba.org for more details.

in a vast number of code repositories. The old structure and the number of systems make it hard for Group A to secure the quality since it raises uncertainties of which code to analyze and where it can be found. These uncertainties make the collection of the meta-data (1) a time consuming activity, thus, without good meta-data the output will likely be invalid. A restructuring to secure the quality of the meta-data and make an automated process possible would need a big investment from the individual division, which is not something that Group A has any direct influence over.

One of the assignments of Group A is to spread best practices in the organization, hence; they have launched initiatives about implementing specific systems in each division to ease their task of monitoring quality. The systems are intended to build a "quality platform" in each division which is used as a baseline for all the code, i.e. all written code is built upon the platform. With a common platform in place, the divisions can better assess their own code-repository and monitor the quality due to customized software rules and standardization. For further information about the quality platform see section 5.1.2. Group B is currently the only division that has a quality platform in place. These systems would indirectly benefit Group A since they would ease the retrieval of the meta-data due to a standardization of the systems. However, the execution of these initiatives depends on the internal priorities and the resources available.

	Group A
Push	• Better meta-data coming from top-management; • Technology initiatives, such as the above mentioned quality platform, towards the divisions to ease the retrieval of quality input for Group A.

Table 2. Push factors identified from the previous section.

Except for the technical-process that is mainly tool driven, the process toward the project-stakeholders is an iterative process that essentially consists of questions and answers. Upon a question, Group A also tries to answer all the surrounding questions that the first question may have raised. By working in this way Group A show what information is available and what is possible with the current technical process. Group A pushes technical reports with key figures, over fixed time periods, towards top management but they also answer requests from stakeholders. Their goal is to be transparent with the collected data and grant access to software metric database by demand.

For future development of their technical process Group A have defined an internal roadmap that lists extensions of their current collecting process and when those extensions should be in place. The roadmap state month-wise time periods when a certain metric (product) should be implemented and collected, e.g.:

- Summer 2011:
    - Percentage of Rule Compliance & Violations Categorization for the Java-code in all the developing divisions.

  - The amount of generated code, the number of lines and files for all languages.
  - Delta KLOC (which describes added, deleted, modified and unchanged code for specific components) for all languages.

- Autumn 2011:
  - Extension of Percentage of Rule Compliance & Violations Categorization to include C and C++ code for all developing divisions.
  - The percentage of duplicated Java, C and C++ code.

In addition, the roadmap includes a list of possible threats and difficulties for future development of the software metric process. Furthermore, there is a continuous project about simplifying and automating the current technical process. The simplifying and the automating steps are piecewise done by members of Group A. From the previous section the following push and pull factors that affect their work with the measurement program have been identified and are displayed in table 3.

	Group A
Push	• *Better meta-data comes from higher priority from top-management;* • *Technology initiatives towards the divisions to ease the retrieval of quality input for Group A;* • Direct access to reports or database for interested stakeholders; • Additional information in the reports, i.e. information regarding the other questions that arose from the original one with the purpose of providing an absolute answer. That additional information show what the current process can or cannot do, hence create incentives for further investment.
Pull	• Answer "why" questions regarding the internal code-environment; • If top management see added value they will invest more in software metrics; • Initiatives to trigger large investments on internal quality can be executed when indicators show ‘red’; • Reports with general software metrics to management, to help them quantify the current situation.

Table 3. Push and pull factors identified from the previous section, the previously identified push factors are marked with italics.

Furthermore, from the results presented above a roadmap, figure 5, has been put together. It aims to graphically describe how contextual factors influence their measurement program. The market needs that affect Group A is mainly requests from top management. When they see added value, and want information that Group A currently cannot provide, they allocate more resources to make it possible for Group A to retrieve that data and refine their process. In addition, currently the metric process is funded on an information need/want basis; hence, if the metrics would indicate low quality on certain areas it would trigger further investments due to the raised information need.

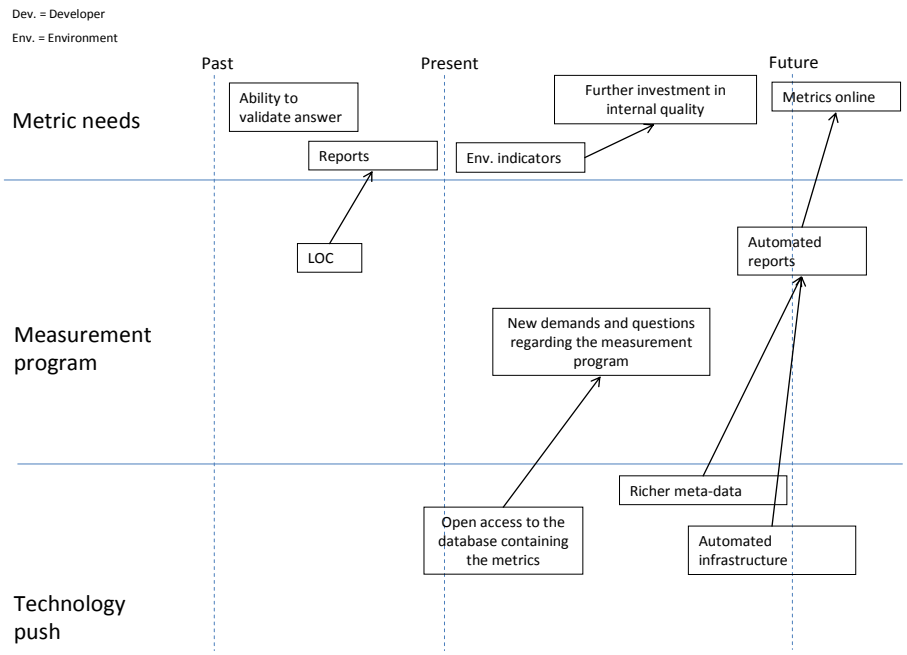

Figure 5. Roadmap over how Group A and their measurement process are affected by contextual factors.

5.1.2 Group B

The process in which Group B collects software metrics is highly tool driven. The main objective for Group B is to support the developers in one of the largest developing divisions of the organization. That implies that it is of their best interest to address the code quality, the developing environment and everything that can affect the developer.

Group B uses two main tools to collect software metrics and ensure the quality of the developing environment. For the collection of software metrics and, to assess the overall code quality they use a commercial tool called Sonar[2]. Sonar analyzes the code on a project basis and displays the results through a dashboard that is accessible to everyone that is interested. The process that collects the code quality related metrics is completely automated and can be seen in figure 6.

[2] Sonar is an open platform to manage code quality. It covers architecture and design, duplications, unit tests, complexity, bugs, coding rules and comments. See www.sonarsource.org for more details.

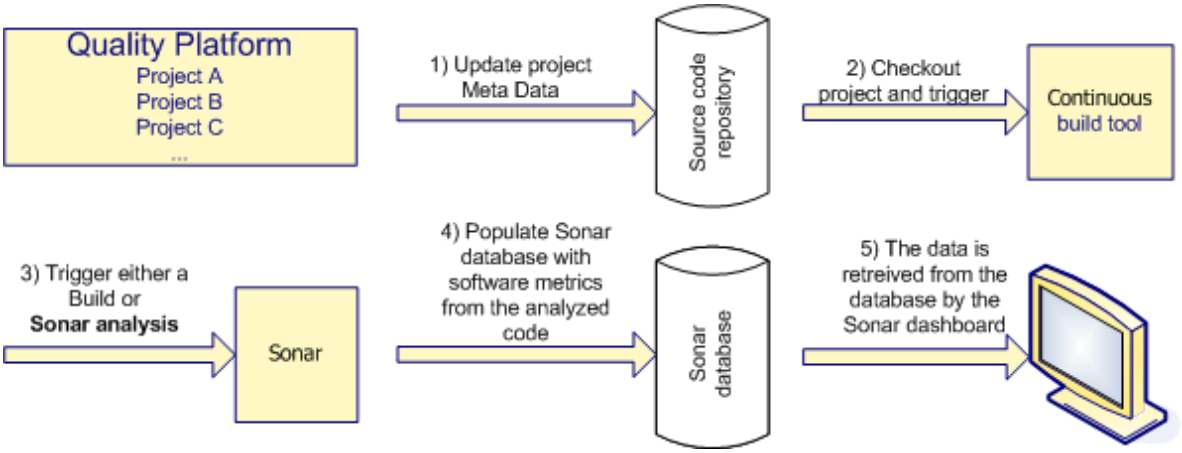

Figure 6. Overview of the data-collecting process used by Group B.

The Sonar process works as follows, (1) a developer checks in a project to the source code repository, on scheduled times (2) a continuous build tool is triggered to checkout code from the repository. (3) The continuous build tool either builds the out-checked code or call Sonar for a metric analysis. (4) When Sonar is called it analyzes the code and populates a database with software metrics. (5) The metrics is retrieved and displayed through a Sonar dashboard for easy access for developers, managers and others.

The key to the above process is the quality platform. In contrast to other divisions within the organization, the division that Group B supports has a defined development environment with a quality platform in place which provides a standardized baseline for the projects. The standardization makes an automated collection process possible. The collection process is designed to be robust and to help project managers to follow the evolution of a project in terms of code quality by using objective measures.

The process of monitoring code quality is continuously enhanced to better suit the divisional needs, and there is planned future enhancement which makes it possible for the developers to monitor which effects their code has on the overall project before they check it in to the repository.

	Group B
Pull	• Code quality assessments; • Sonar dashboard.

Table 4. Pull factors identified from the previous section

The other tool that Group B uses is developed in-house, hereafter referred to as Devtool. Devtool was designed with the purpose to make it easier for developers with their day-to-day activities and Group B took responsibility over the tool mid-2010 and invoked statistic logging to the tool. When a developer uses the tool, all the information about the activities is stored into a dedicated database maintained by Group B. The main incentive behind the tool is, as previously mentioned; to make it easier for the developers but Devtool also collects statistics that make it possible for Group B to better support the developing environment. Today Devtool is mainly

used to monitor the size of the tools user-base, to give Group B an indication if it is worth continuing to invest in. (For now Devtool is used by roughly 20% of the developers in the division.) With the statistics, from the development environment and its surrounding, Group B can get a better perspective about which parts need improvement and early spot possible degradations in the environment. A sample of the type of statistics that Devtool collects can be seen below in table 5. This statistic is specific for this particular organization and would not be suited in a different organizational context.

Action (Install/build/etc.)
Date and time
Execution/Duration (time in seconds to complete the action)
Product (the product the user is working on)
User (the user that execute the action)
Release (which release of the product)
Message (outcome of action Success/Error)
Complexity of action (nr of components used)
Where the action was called (remote/local)

Table 5. Sample parameter statistics from Devtool which makes it possible for Group B to detect internal degradation and take appropriate action.

Since Group B is mainly concentrated on supporting the developers they do not have any formal process in place regarding reporting. Conversely, Group B does collect a lot of data and the data collecting is a high priority, additionally, Group B can compile reports to interested stakeholders. However, regular reports and business intelligence activities are not a priority by the division and are only done in isolated cases upon request.

From the previous section the following push and pull factors, that influence Group Bs work with their measurement program, have been identified and are displayed in table 6.

	Group B

Push	• Statistic logging for Devtool;
	• Increase the use of Devtool by developers in all the developer divisions;
	• The benefits of statistics that come from the use of Devtool to top management, statistics such as:
	  ○ The duration it takes to complete certain actions for a specific release of a product, which makes it possible to spot degradation in the workstations, anomalies between the releases, etc.;
	  ○ The percentage of successful/erroneous outcomes of an action for a specific release of a product, if it is a high error-percentage they have the ability to drill down and correct potential defects;
	  ○ If the effort of componentization pays off, i.e. if the developers always build a “full view” or if they rather build with a fixed number of components.
	• PP-presentations (Microsoft Power Point) of key figures;
	• Reporting actions to provide top-management with reports. (Note, top-management does not pull these reports, it is more an effort from Group B to push reports to

	them to show what data they have in order to gain recognition.)
Pull	• *Code quality assessments*; • *Sonar dashboard*; • Developer committees want Group B to collect data (Devtool, Sonar) ; • Graphs and statistics regarding the status for the development environment; • Better monitor the development environment. ○ To early spot possible degradation and maintain good functionality.

Table 6. Push and pull factors identified from the previous section, the previously identified pull factors are marked with italics.

Furthermore, from the results presented above a roadmap, figure 7, has been derived and aim to graphically describe how contextual factors influence Group Bs measurement program.

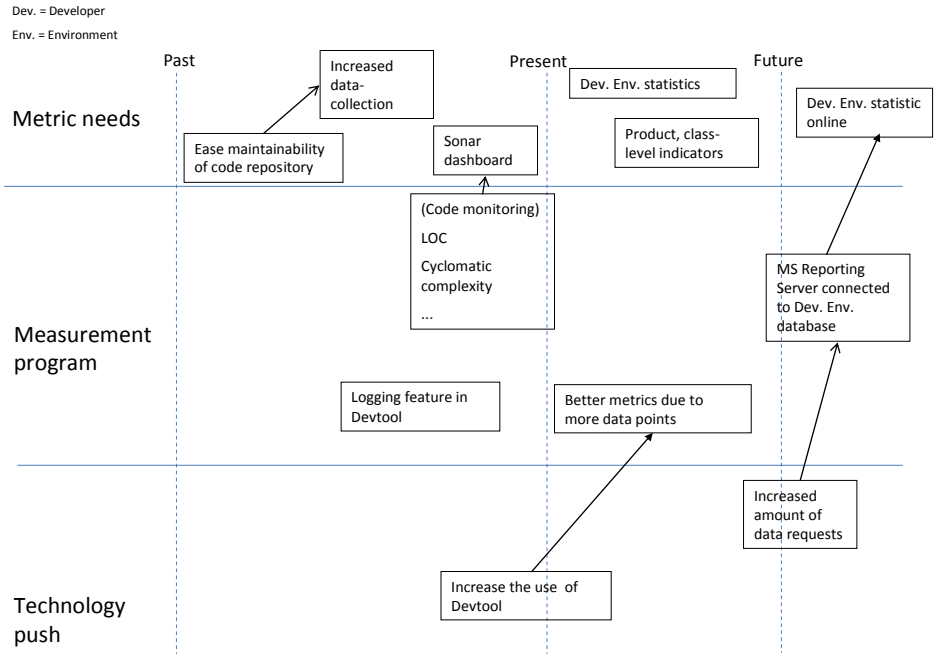

Figure 7. Roadmap over how Group B and their measurement process are affected by contextual factors.

5.2 Product

This section presents the results from the data collection regarding the 'Product' and the product of the respective processes will be presented in a table and is categorized by how it conforms to the definitions given by the ISO 9000 standard.

5.2.1 Group A

The table displays which software metrics are collected and currently used. The metrics is calculated from the raw code by the tool Resource Standard Metrics[3] (RSM). Group A aims to keep the metrics as basic as possible until the technical process is more mature. One example is the calculation of functional points (FP), the FPs are directly derived from the lines of code based on the recommendations from Quantitative Software Management (QSM) 2009 [24].

Group A	**Functionality**	**Reliability**	**Usability**	**Efficiency**	**Maintainability**	**Portability**	**SUM:**
Quantitative metrics							
Number of statements	1	1	1	1	1	1	100,0%
Number of comments	0	0	1	1	1	1	66,7%
Number of files	1	1	1	1	1	1	100,0%
Lines of code (LOC)	1	1	1	1	1	1	100,0%
Cyclomatic Complexity	1	1	1	1	1	1	100,0%
Functional Points (FP)	1	1	1	1	1	1	100,0%

Table 7. The main metrics that is collected and used by Group A and their conformance to the ISO 9000 standard.

The metrics above are mostly general, base-metrics, and not as specialized towards a certain context. That is since Group A have an outspoken policy to start slow and build from that, i.e. no advanced metrics that may be misinterpreted. Also, they cannot be completely context specific due to the fact that they serve the whole organization.

The target audience for the reports that Group A generate is mainly interested in high level figures, such as the overall code status from where they can drill down deeper if it would be necessary. However, the collected metrics are used to compare the different divisions, on a language basis, on how many LOC, FPs, number of files, etc. they have and put that in relation to the organizations code repository. From there they can overview which division that is largest, from a source code perspective, and which languages are used in the organization. By doing this continuously they can see how their corrective efforts are progressing, e.g. try to minimize the amount of legacy code.

The roadmap that was discussed in section 5.1.1 also states which new metrics (products) that Group A will start to collect and when the implementation is complete, i.e.:

- Summer of 2011:

[3] Resource Standard Metrics is a source code metrics and quality analysis tool which provides a standard method for analyzing C, ANSI C++, C# and Java source code across operating systems.

- Delta KLOC.
- The amount of generated code for all languages.
- The percentage of rule compliance, for java code.
- The number of violations, for java code.

- Autumn of 2011:

  - The percentage of duplicated Java code.
  - The percentage of duplicated C code.
  - The percentage of duplicated C++ code.
  - The percentage of rule compliance, for C code.
  - The percentage of rule compliance, for C++ code.
  - The number of violations, for C code.
  - The number of violations, for C++ code.

5.2.2 Group B

Table 8 below displays the output (product) from the technical process regarding the tool Sonar. The displayed metrics are the ones that are mostly used; however, Sonar derives a plethora of metrics depending on which plug-ins that are implemented (for a complete list see [25]).

Group B	**Functionality**	**Reliability**	**Usability**	**Efficiency**	**Maintainability**	**Portability**	**SUM:**
Quantitative metrics							
LOC	1	1	1	1	1	1	100.0%
Number of comments	0	0	1	1	1	1	66.7%
Duplicated code	1	1	1	1	1	1	100.0%
Number of classes	1	1	1	1	1	1	100.0%
Number of code violations	1	1	1	1	1	1	100.0%
Cyclomatic Complexity	1	1	1	1	1	1	100.0%
Rules compliance	1	1	1	0	1	1	83.3%
Code coverage	1	1	0	0	1	1	66.7%
Test success percentage	1	1	0	0	1	1	66.7%

Table 8. The main metrics that currently is collected and used by Group B and their conformance to the ISO 9000 standard.

The interesting part, from table 8, is that these metrics differ from the ones collected by Group A in the sense that they are more specialized towards their context and not as general as the metrics collected by Group A.

5.3 Descriptive statistics, roadmap and important aspects of software measurement processes

This section presents the descriptive statistics derived from the quantitative data collection, the final roadmap and a table with three identified key factors for

efficient data collection, aimed towards organizations that work under similar contextual factors.

Table 9 displays descriptive statistics from the quantitative data collection with the purpose of assessing how Group A and Group B perceive their current situation. Also, to highlight which areas that they need to further address to improve their measurement process. Table 9 show the percentage of conformance with the questions that can be found in table 1, the values denotes the mean answer from the sample where a 3, on the 0-3 scale, represent 100%.

Question	**Group A**	**Group B**
1	56%	56%
2	22%	22%
3	11%	0%
4	67%	33%
5	67%	44%
6	67%	44%
7	78%	78%
8	78%	56%
9	78%	89%
10	67%	33%
11	78%	33%
12	67%	56%
13	44%	33%
14	44%	11%
15	56%	67%
16	33%	44%
17	56%	33%
18	22%	11%
19	44%	11%
20	78%	56%

Table 9. The degree of conformance with the questions in Table 1. The interesting fact is how big the difference is between these groups on questions 11, question 14, etc. and highlights which factors that vary in these two contexts. .

In the above table we can observe that the main discrepancies between the groups are the parts that concern *product* and *context* (question 10-20). One particularly interesting part is question 14, *"Are the collected metrics used to its full extent, i.e. are all the collected metrics used as support for some decision(s)?",* where the answer from both groups is in the lower half, and Group B as low as 11%. In general, the conclusions that can be drawn from the above table are that on many points (questions) the two groups identify their situation as more or less equal. On the other hand, the points (questions) that show a big discrepancies illustrates on which points there is a contextual difference between the groups, which could be used for further analysis. (However, such analysis will not be covered in this study.)

In addition, to further highlight the difference in perception between the two groups table 10 display the calculated CSI value, i.e. the value of the overall tendency to answer '*Yes, completely'* or *'Yes, almost completely'.* These percentages should be

interpreted as the total group compliance with the best-practice based questionnaire.

Upper half (2 or 3) frequency:	
Group A	67%
Group B	40%

Table 10. The perceived group satisfaction with their current process derived from the questions in Table 1.

From table 10 we observe that Group A have a considerably higher satisfaction rate than Group B, something that might strive from the fact that Group A work directly with top management and are more dedicated to their software metric process. On the other hand, Group B acts as a service team that primarily use their software metric processes as a mean to serve the division, which could be one possible explanation for the lower satisfaction rate. It can be argued that the objective for Group As work is their software measurement process, in the meantime, Group B use their software measurement process as a tool to fulfill another objective, i.e. a better divisional development environment. By using the measurement process as a tool implies, in this case, more constraints and less recognition from external stakeholders. Therefore, it affects Group B in the sense that they do a lot of work but they do not get the same recognition as Group A due to the fact that their work and effort only is evident internally within the group.

The above mentioned statistic is meant to provide further insight and background information to the roadmap (figure 8) that has been derived to illustrate and answer the first research question:

- *How two software measurement processes within an organization that already works with software metrics are affected by contextual factors?*

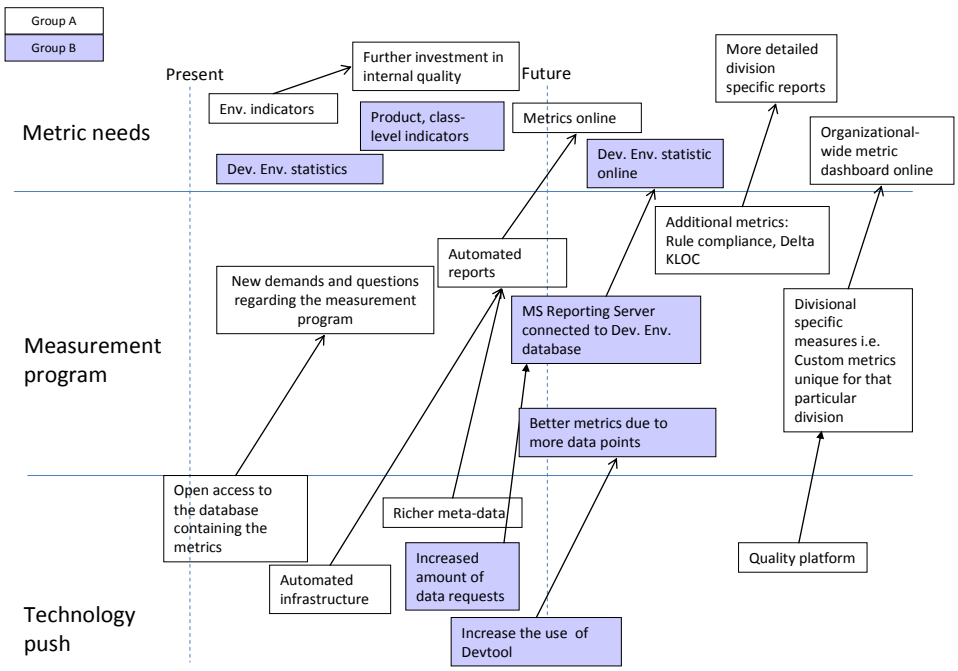

Figure 8. Interlaced roadmap describing how Group A and Group B currently are affected by contextual factors, their future initiatives and future drivers for a more efficient data collecting process.

The roadmap is based on the findings in section 5.1 and 5.2; it is interlaced to provide a concurrent picture of the current status and future initiatives and consequences for the measurement processes in the organization. From the roadmap it is possible to identify variations, and similarities, between Group A and Group B, e.g.:

i. That Group A needs substantial support and funding to be able to extend their measurement program based on the technology push factors.
ii. That Group B, since they mainly are a service team, has far less initiatives planned than Group A.
iii. Neither of the two groups have long term plans for their measurement processes.
iv. That the main driver (market need) for both groups are the essentially the same, i.e. provide the market with environmental indicators, preferably online.

A possible reason for (i) is the fact that Group A has more ambitious initiatives planned than Group B. Since, Group A is dependent on all the other divisions in the organization in order to get their initiatives realized, which is something that Group B does not need to take into consideration since they only work internally in their division. The fact that Group B mainly is a service team can be concluded as the explanation for (ii) due to the fact that they use the measurement program as a tool to provide a better development environment. Thus, their main goal is not the software measurement process since that process is refined piecewise on a need-basis. Hence, there are no future initiatives in place since the tool (process) is adapted on the basis of the objective, which is a better developing environment. Furthermore, an explanation of (iii) is as both groups have limited funding, one based on management's willingness to invest in internal quality and one moderated by the division and its priorities, no one of the two groups can have any particular long term plans for the measurement process, since there are no dedicated resources. However, the contextual factors points out that the two groups have a lot in common, they both strives towards (iv) and it can be argued that they could gain a lot by raised communication between the groups. That is since the main driver of the measurement-processes is the same, with a micro or macro detail, and communication and collaboration would make it possible to make use of potential synergy effects. Additionally, both groups strive for an automated process and easier access to metrics, and where the one group fall short the other group excels, e.g. Group B has a sophisticated collecting process and a quality platform in place but no real external support or recognition, on the other hand, Group A has a slightly lacking process but a close collaboration and support from key stakeholders. Hence, there are a lot of beneficial unifying initiatives that can be made concerning the two groups.

In addition, three important factors for efficient measurement programs have been elicited. The purpose for these factors is to further cover the contextual factors that

affect a measurement program, and to provide concrete information about how these factors contribute for the long term success of a measurement program. Hence, answering the second research question:

- *How do internal pull factors contribute to assuring efficient data collection in the long run?*

The stated factors are listed, along with descriptions of why each factor is important for respectively group, in table 11.

	Group A	Group B
Support	• Group A need long term support from management to ensure that their software metric program assimilates to the organization, a feature also pointed out by [2]. • Long term support and higher internal priority is key to raise the respect for software metrics, which only 44% (table 9, question 13) think they have now, and increase the internal communication of the software metric program.	• Support is an important part of a successful and efficient software metric program as mentioned by [2] and [11]. Currently, only 33% (table 9, question 4) think their data-collection get enough support from the organization.
Definition	• If Group A would define their processes by working according to a standard such as ISO/IEC 15939 it would increase their process transparency. In addition, a clear process definition would reduce the risk of interpretation errors which is important for a successful measurement program [12, 15]. • Also, a more defined process leads to a less people	• Group B would benefit from being more precise and clear with their current collecting process, e.g. [16] concludes that metrics only can be collected in a concise manner if the data collection follows a predefined company-wide procedure. • Also, Group B would benefit from having clearly defined

	dependent process [12] which would be beneficial since there is no available software metrics training at the organization.	customers for the collected metrics. Defined customers is important, part for support and part for the fact that the customers are the ones that will make decisions based up on the collected metric [10].
Refinement	• Group A need, with the help of top management, to push the divisions that have old legacy systems towards a restructuring. To ease data retrieval and automate the collection process, which is essential to becoming more efficient and successful regarding software metrics [12, 14]. • Furthermore, refine the process by always having a clear customer for the collected metric to ensure that the metrics are being used in decision making, which is highly important for a successful measurement program [9]. In addition, used metrics would increase the chances to spot anomalies in the collected data [10, 12].	• Extend the current process by invoking reporting actions. Westfall [10] stress the importance of having reports connected to the data collection. Otherwise, the chance is that the data only is collected for the sake of collecting (which 33% currently thinks (table 9, question 13)).

Table 11. Three highly important aspects for efficient metric collection, elicited from the contexts in this case-study.

6 Validity evaluation

The threats and uncertainties concerning this study are identified using the categories presented by [26]. Thus, the main threat for the external validity in this study is that it is only covering one organization. However, the key criteria that was elicited from both objects relates well to best practices identified from the literature. Also, even though these objects work in different organizational contexts, they shared the same important aspects for a more efficient data collection.

The central threat towards the construct validity is the fact that this case study was done under a mono-operation bias. Hence, the objects where only studied under a short single period of time, which may present a result that is only valid under just that period of time. But, since the processes are no older than a 1-2 years it can be argued that the current results are valid for the complete history of these processes, since there have been no signs of process degradation.

The major implication with the internal validity is the selection of the candidates that were interviewed. Even though the selected candidates did possess the adequate knowledge, their answers could have been personally biased due to their current situation. Nonetheless, the objective was to investigate how contextual push and pull factors affect them and the personal bias could be interpreted as a result from those factors.

Regarding the conclusion validity the main threats are that the sample size from the quantitative data collection was too small for any formal statistics and that the questionnaire was untested. However, no one of the two groups have more than three to five dedicated members and it was three respondents from each group for the questionnaire. The questionnaire was designed to represent a loosely best-practice scenario, with the purpose of quantify how well their current situation conformed to best practices within the subject. In addition, the roles within the groups did cover different responsibilities, which could have affected their personal view of the questionnaire.

7 Conclusions and Future work

It is difficult to obtain maximum value from software measurement programs since they can be executed and used in several ways, and it is not always possible to say if the collected metrics actually are used or just collected. Hence, an important aspect regarding measurement programs is the purpose, to be able to answer *why* the data is collected. An underlying purpose is important for minimizing the chance of a program write-off, something which there is many examples of in the literature, when the sole reason behind the program is that others in the industry are doing the same thing. A replicate approach often leads to the devise “technology for technology’s sake” [27] which is ill-suited for the longevity of a measurement program.

The constructed roadmap illustrates the current state, and future possibilities, for the studied organization. It also shows factors that strive from technological actions

and how they relate to each other. The next key step for the organization, if they want to take their measurement programs to the next level, is to spread awareness of the software metrics and create an incentive program around the metrics. The purpose would be to raise the internal respect and awareness for software metrics and stimulate developers, managers and other stakeholders to use the available metrics for decisions. From there they can start to refine the measurement processes by defining clear customers (from raised awareness) of the collected metrics and gain further support from the organization.

The presented roadmap is by no means applicable as a general description for contextual factors in all organizations that have a software measurement program in place. Moreover, the findings in this report do not serve as complete guidelines for organizations that want to be more efficient and long term with their data collection. Rather, the findings in this report should be used as a baseline when analyzing the inner workings of an organization that want to assess and improve their software measurement program. By using roadmapping for internal analysis and ensure that the three key factors are met, organizations can secure their measurement process and assess internal areas for improvement, to guarantee a more efficient and long term data collection.

This study is based on a period of five months working at the organization. Thus, this study draws, to an extent, on anecdotal evidence gained from the time at the organization and is partially influenced by the environment and sightings during that time. However, this is an effort to help organizations develop their existing measurement processes and to make them more efficient, hence, gain more value from them.

Suggestions for future work would be to practically develop a software measurement program in a real, software intense, organization and analyze:

- Which software metrics that generally can be categorized as “relevant metrics”? Contrastingly, which metrics that seldom can be categorized as “relevant metrics”?
- Political factors, how does the organizational politics affect the measurement program and why?
- Deep behavioral analysis on the developers with the purpose of assessing why they tend to be resilient to measurement programs and why?

The factors are interesting for providing a baseline for future development of software measurement programs. To analyze which tangibles (metrics) that is most important and which intangibles (politics, resilience within the organization) that should be addressed to prevent that the organization hinders itself from success.

8 Acknowledgements

This project has been done in parallel with an internship at Amadeus site in Nice, France. The author would like to thank all the Amadeus-personnel that made this report possible, especially Dirk Ettelt and Christophe Vallet. In addition, the author

would like to thank his supervisor, Miroslaw Staron, for his invaluable feedback and guidance through this project.

9 References

1. Reel, J.S., *Critical success factors in software projects.* Software, IEEE 1999. 16(3).
2. Gopal, A., T. Mukhopadhyay, and M.S. Krishnan, *The impact of institutional forces on software metrics programs.* Software Engineering, IEEE 2005. 31(8).
3. Limited, D.M., *IT Projects: Experience Certainty*, T.C. Services, Editor. 2007.
4. Jones, C., ed. *Applied Software Measurement: Global Analysis of Productivity and Quality* 3ed. 2008, Osborne/McGraw-Hill. 662.
5. Bom, B., *Software process improvement: biting the bullet*, in *Innovation in Technology Management - The Key to Global leadership. PICMET '97: Portland International Conference on Management and Technology* 1997: Portland, OR , USA.
6. Fenton, N., *Software Metrics: Successes, Failures, and New Directions*, in *SM/ASM*. 1999.
7. DeMarco, T., *Controlling Software Projects: Management, Measurement, and Estimates*. 1986: Prentice Hall. 296.
8. Rubin, H., *Measuring 'Rigor' and Putting Measurement into Action.* 1991.
9. Kitchenham, B., S.L. Pfleeger, and N. Fenton, *Towards a framework for software measurement validation.* IEEE Transactions on Software Engineering, 1995. 21(12).
10. Westfall, L., *12 Steps to Useful Software Metrics*. 2005: Plano, TX.
11. Staron, M., W. Meding, and C. Nilsson, *A framework for developing measurement systems and its industrial evaluation.* Information and Software Technology, 2008. 51(4).
12. Staron, M. and W. Meding, *Factors Determining Long-term Success of a Measurement Program: An Industrial Case Study.* e-Informatica Software Engineering Journal, 2009. 3(1).
13. Gopal, A., et al., *Measurement Programs in Software Development: Determinants of Success.* IEEE Transactions on Software Engineering, 2002. 28(9).
14. Seaman, C., M. Umarji, and H. Emurian. *Acceptance Issues in Metrics Program Implementation*. in *METRICS '05 Proceedings of the 11th IEEE International Software Metrics Symposium*. 2005. Washington, DC, USA.
15. Herbsleb, J.D. and R.E. Grinter, *Conceptual Simplicity Meets Organizational Complexity: Case Study of a Corporate Metrics Program*, in *20th International Conference on Software Engineering (ICSE'98)*. 1998: Kyoto, Japan.
16. Panfilis, S.D., B. Kitchenham, and N. Morfuni, *Experiences introducing a measurement program.* Information and Software Technology, 1997. 39(11): p. 745-754.
17. Niessinka, F. and H.v. Vliet, *Measurement program success factors revisited.* Information and Software Technology, 2001. 43(10).

18. Solingen, R.V. and E. Berghout, *Goal/Question/Metric Method: A Practical Guide for Quality Improvement of Software Development*. 1999: McGraw-Hill Inc. 280.
19. Basili, V., G. Caldiera, and H.D. Rombach, *The Goal Question Metric Approach.* 1994.
20. Phaal, R., C.J.P. Farrukh, and D.R. Probert, *Technology roadmapping - A planning framework for evolution and revolution.* Technol. Forecast. Soc. Chang., 2004. 71(1-2): p. 5-26.
21. Seaman, C.B., *Qualitative Methods in Empirical Studies of Software Engineering.* IEEE Transactions on Software Engineering, 1999. 25(4).
22. Lethbridge, T.C., S.E. Sim, and J. Singer, *Studying software engineers: Data collection techniques for software field studies.* Empirical Software Engineering, 2005. 10: p. 311-341.
23. R.Jeffery and M.Berry, *A framework for evaluation and prediction of metrics program success.* 1993.
24. Quantitative Software Management, I. *Function Point Languages Table*. 2009 [cited 2011 9 April]; Available from: http://www.qsm.com/?q=resources/function-point-languages-table/index.html.
25. scmGalaxy. *Compare between RSM and Sonar*. 2010 [cited 2011 May 05]; Available from: http://www.scmgalaxy.com/sonar/compare-between-rsm-and-sonar.html.
26. Adams, D.J.A., *Statistical Validity Pitfalls*. 2008, Vanderbilt University: Nashville, TN USA.
27. Bensaou, M. and M. Earl, *Right Mind-Set for Managing Information Technology.* Harvard Business Review, 1998.

Appendix
This section contains all the documents that have been used through the report.

Tabulation

	Group A	Group B
Process: Past	Started in the end of 2009 with a System planning group that was providing statistics ad-hoc statistics (e.g. lines of code (LOC)) to top management. However, it presented a few problems, the results completely relied on the input from the divisions, no formal database to store the results in and there was a big variance that made it hard to rely on the results This process was stopped and CSE started in the end of 2009. Main goal - to be efficient we have to provide more data than LOC since there is a lot of ways to challenge the results with just LOC. We needed to go further,	Devtool was developed to help developers with their day-to-day activities. Devtool makes is possible to see if the tool is used or not.

	for instance if we should estimate the effort we can't be happy with just LOC. The new metric program started as a way to answer a simple question from top-management. Then it went on, little by little since we needed to structure our answers to the questions. The metric process was developed through an iterative process, questions, and answers that pushed for something more.	
Current	Database in place for processing the findings within the divisions. The findings are meta-data provided by the code-repositories in the organization. We synchronize that source code with our repository then we do all the counting and store the results in our database. All the quantitative metrics is in place but we don't know yet about the quality metrics. We (Group A) are very attentive to the questions divisions/departments ask - they show their requirements through their questions and that is why we present the metrics for the divisions, to collect feedback because that is the way to feed our work. We (Group A) provides the same metrics to all departments. Metrics that will defer when we have the quality-platforms in place since it is different languages, rules, etc. Different departments have different maturity towards metrics that we have to adapt to. We propose new metrics to departments but in the same time we exchange difficulties with them to get the metrics stable for the long term. There are still manual steps in the process that we have to automate, also to communicate the importance of providing us with good meta-data. On	The purpose of this team is to drive the builds, tests, and everything else around source control. Also, everything around supporting the developer, e.g. to monitor if the tools provided to developers are good enough and used. We use the statistic for monitoring the acceptance of the tool (Devtool). If it is accepted or not, if it is used or not? Originally developed by someone in [another developer division] and we took the leadership of this tool and worked on it since July 2010. In the beginning only used by (another team in Amadeus) now it is used by several teams within Amadeus. Devtool collect a lot of things but for now it is only used to monitor the acceptance (of the tool itself). We always try to enhance Devtool so it corresponds with user needs. E.g. if one manager is not interested in benchmark data and he/she wants to know what is really happening on real developer machines then Devtool could help with this

	certain areas the development process is not that precise hence, it is hard when we ask for the meta-data since some divisions are not managing their code repository, as well as they could, and the directory is not as precise as it could be. We know that we have to adapt. When we ask for the meta-data, we are suffering from the divisions internal priorities. They are not always ready to provide us with the input that we would need. We use an iterative process, Q&A, push and pull relationship.	problem; it should be used more in this sense of what is happening on developer machines. Probably more costly for Amadeus if every division developed their own tools. For now statistics is not known outside of this team. This team (Group B) is for providing support for developers, this team is naturally a team that collects data and we should provide reports to management. To monitor the effectiveness of the teams since we have the data, for example in Sonar, which we could generate BI-reports from. A lot of data but no reports to extract.
Future	We need to investigate what is happening with the code, e.g. lines of modified/created/deleted LOC (as stated in the roadmap) and make sure that we are improving. We want to establish a quality platform in each division. Where it is up to the division to define all the rules and violations that they want to detect. The platform will be managed by the division and we will only set up a couple of rules that we will manage in the central code repository. However, before implementation we need to see what added value we can give and how we should proceed with the project, it will be an iterative process. In addition, we (Group A) want to put in place benchmarking against an industrial reference. We strive towards an automated process regarding metrics and to be transparent with the metrics.	Increase the number of users of Devtool, today there is maybe 20% of the developers using it. On the other hand, if the developers do not like this tool, we will not use it anymore.

	Quality figures for 2011 will decide if they (divisions, top management) will invest in the quality platform. They invest if they think it will generate a good return of investment.	
Product:	Our goal is to have the code for all the java-projects in the organization. The purpose is to see the overall code-quality and detect code that should not be allowed into production. We push validated reports of for a given time-period with general code-statistics (LOC, FP, etc.) to top-management. We profit when we present the results, but not only the results since we are also explaining the process of collecting. It provides them with status reports to assist with decisions where to put their money, e.g. to keep investing in the metric-program.	The purpose of this team is to drive the builds, tests, and everything else around source control. The purpose is to, since we collect several things, know how long time it takes to compile/install/etc. and there are a lot of statistics that can be derived from this. In addition, we have statistics about build time, failed builds, etc., all data that comes from using Jenkins/Hudson (http://hudson-ci.org/). Devtool is different since we developed it; we collect data and generate reports. For now we derive statistics regarding the number of developers using Devtool. With Devtool we could generate a performance graph per workstation, product and release. We want to monitor the weekly performance for each machine then we can detect if there is trouble with deliverables. We should provide reports to management for monitoring the effectiveness of the teams. However, Devtool is not for business reports, it is for internal development and similar issues.
Performance:	We provide decision support that comes from being able to quantify the current situation and raised awareness on points that can be improved by top management.	Devtool provides monitoring if it is worth investing in this kind of tools, measure the ROI which have been good this far

	Also, when management talk they know the figures and can put it into relation with industry standard, etc.. The process brings added value due to raised awareness and support.	We have different topics to deliver and improve the build-time, quality of the development environment and benchmark workstations. We suggest replacing workstations more often, and the reason behind that action is the question “are we providing sufficient hardware to people?” There is a lot of complex task at hand, some of them Devtool should help with. We have to find a way to show them that they (the developers) can save time by using Devtool.
Key areas:	We have to provide management with metrics they can use, it is really important when you present a framework for the metrics to know what questions they might have and be able to answer them. Also to keep the information relevant, otherwise they will not be interested. Be able to answer "why" questions regarding code. Need good meta-data and an automated process to be successful. Also, support since without management support it will be no accessible data. If there is any new data directors want to see they know that we need their support. Better meta-data comes from higher priority from top-management, hence, if they see added value they will invest more. The primary priority from other divisions is to have their division up and running, how they chose to do that is up to them. They have to provide us with data but the production will always be priority one. Our questions are important but we have to be flexible and adapt.	A new investment comes from the user-base of a certain action We are a service team, with few employees, that services a big part of the organization; we can invest in one tool, not ten. If other divisions want to invest in something else, they can. The objective for the other divisions is to deliver their product on time. As long as they do that they can to whatever they want. Hence, it is up to us to show them the gain by using Devtool. We want to focus on Devtool and show management that Devtool collects statistics. Hence, if they want statistics we should use Devtool. Also there are requests that we should collect data but reporting is not a top priority. We have the data but we never export the data. Possible to say that every division is closed down (towards the others) and is its own sub-company.

		Having the data is one thing and having the report is another, better, thing. We need to target actions associated to a report.

Table 12. Analyzed data from the qualitative interviews, the data is formatted to the most essential findings in each "code" category.

Cross tabulation of the questionnaires

	Group A					Group B					A + B					
	0	**1**	**2**	**3**	**N/A**	**0**	**1**	**2**	**3**	**N/A**	**0**	**1**	**2**	**3**	**N/A**	**Sum**
1	0%	33%	67%	0%	0%	0%	67%	0%	33%	0%	0%	50%	33%	17%	0%	100%
2	67%	0%	33%	0%	0%	33%	67%	0%	0%	0%	50%	33%	17%	0%	0%	100%
3	67%	33%	0%	0%	0%	100%	0%	0%	0%	0%	83%	17%	0%	0%	0%	100%
4	0%	33%	33%	33%	0%	33%	33%	33%	0%	0%	17%	33%	33%	17%	0%	100%
5	0%	0%	100%	0%	0%	0%	67%	33%	0%	0%	0%	33%	67%	0%	0%	100%
6	0%	0%	100%	0%	0%	33%	0%	67%	0%	0%	17%	0%	83%	0%	0%	100%
7	0%	0%	67%	33%	0%	0%	0%	67%	33%	0%	0%	0%	67%	33%	0%	100%
8	0%	0%	67%	33%	0%	0%	33%	67%	0%	0%	0%	17%	67%	17%	0%	100%
9	0%	0%	67%	33%	0%	0%	0%	33%	67%	0%	0%	0%	50%	50%	0%	100%
10	0%	33%	33%	33%	0%	67%	0%	0%	33%	0%	33%	17%	17%	33%	0%	100%
11	0%	33%	0%	67%	0%	33%	33%	33%	0%	0%	17%	33%	17%	33%	0%	100%
12	0%	33%	33%	33%	0%	0%	67%	0%	33%	0%	0%	50%	17%	33%	0%	100%
13	33%	0%	67%	0%	0%	33%	33%	33%	0%	0%	33%	17%	50%	0%	0%	100%
14	33%	0%	67%	0%	0%	67%	33%	0%	0%	0%	50%	17%	33%	0%	0%	100%
15	33%	0%	33%	33%	0%	0%	33%	33%	33%	0%	17%	17%	33%	33%	0%	100%
16	0%	33%	33%	0%	33%	33%	0%	67%	0%	0%	17%	17%	50%	0%	17%	100%
17	0%	33%	67%	0%	0%	33%	33%	33%	0%	0%	17%	33%	50%	0%	0%	100%
18	67%	0%	33%	0%	0%	33%	33%	0%	0%	33%	50%	17%	17%	0%	17%	100%
19	0%	67%	33%	0%	0%	67%	33%	0%	0%	0%	33%	50%	17%	0%	0%	100%
20	0%	0%	67%	33%	0%	33%	0%	33%	33%	0%	17%	0%	50%	33%	0%	100%

Table 13. Response data from the quantitative data collection, it highlight the amount of respondents that selected each alternative for each question.

Chalmers University of Technology
Kristian Mattsson mattssok@student.chalmers.se

Transcript Group A

Date: 18/3 – 2011:
Codes:

1. *Process – concern the way that they are working with metrics*
    a. *Past*
    b. *Current*
    c. *Future*
2. *Product – concerns the actual results from the metric work. The end product that is delivered.*
3. *Performance – How are these metrics helping the organization today and how are the used, respected or not.*
4. *Key areas – areas that are critical for the continued work with the metric program.*

K: Kristian Mattsson
TLA: Team Leader for Group A

K: Which year did you start with software metrics?
TLA: We started in the end of 2009. (1a)

K: So the Product Development and Strategy (PDS) department did not have anything before that?
TLA: It was system planning, they were providing statistics to top managements, ad-hoc statistics that were provided due to some goal from the top management. It was regular reports of lines of code. But they did present a few problems. The results completely relied on the input from the divisions. It where no formal database to store the results and the third problem where that there was a big variance and it was hard to rely on the results. This process was stopped for one year and then we started in the end of 2009. The main thought was, to be efficient; we have to provide more data than LOC (1a). Also, we had to own the counting, automated the steps and store the data in our own database. This is something that we put in place in 2010, we have the DB in place and we have to process the findings within the divisions, they provide meta-data from the code-repositories. We synchronize that source code with our repository then we do all the counting and store the results in our database (1b). This is difficult but we want to go further. We also want to investigate what is happening with the code e.g. lines of modified/created/delete LOC (1c).

K: Ok, so you don't see any further then September 2011?
TLA: No. In September there are big items, without metrics description. All the quantitative metrics is in place but we don't know yet about the quality (1b). We want to establish a quality platform in each division. Where it is up to the division to define all the rules and violations that they want to detect. The platform will be managed by the division and we will only set up a couple of rules that we will manage in the central repository. This is something that is really useful for the developers and divisions (1c). For top management it is interesting to know the

evolution of the code but not in detail (3-4). We present a roadmap to the divisions, and we have this data in place.

K: I assume that the main purpose for your metric-collection is to provide management with decision-support, so they can take better business-decisions. And do you feel that the current metric-process has respect from management, that they trust the metrics and use them when they take decisions? Or is the metrics just collected but never used?
TLA: That was the case before, but the feedback where that; if you only present the LOC there is a lot of ways that you can challenge that result. That's why I decided to go further, for instance if we should estimate the effort we can't be happy with just LOC (1a). We had to go further in the analysis to gain respect from the divisions, with respect they will use the metrics as input for decisions. We have to go further in the analysis to take the metrics into consideration (1b, 4). For now when we show the metrics (2), we can see which questions the management will have and we have to be able to answer with another metrics (4). Support the answers and finding with additional statistic. We have to provide management with metrics they can use, it is really important when you present a framework for the metrics to know what questions they might have and be able to answer them, and that the information is relevant otherwise they will not be interested (4).

K: So when you construct the metrics, data-points to collect, are you reverse-engineering them from the questions they might have?
TLA: We have a list of things that is logical and that we can provide (1b). But we have to be very attentive to the questions they ask since they show their requirements through their questions. That's because we present the metrics for the divisions, to collect feedback because that is the way to feed our work (1b).

K: Do you provide metrics to all departments, or just the SEP?
TLA: All departments (1b).

K: Do the metrics you provide to SEP differ from those to e.g. AIR?
TLA: No it is the same metrics. They will defer when we have the quality-platforms in place since it is different languages, rules, etc. (1b).

K: Would you say that it is base-metrics that you are collecting now?
TLA: The metrics now are the same for all (1b).
K: Do the departments have a deadline for implementing of the quality-platforms?
TLA: Yes and no, we have to adapt to the current maturity of the department. E.g. the SEP department has a different history. They use java and got a lot of open-source tools for quality. They already have set up a java common platform (JCP) for quality in SEP. They have already come a long way regarding maturity, they are very good. On the other hand, the central system division does not have any open-source quality platform to use and are interested of implementing the quality platform for a single component to see if PDS can prove added value regarding quality for them. If we prove added value for the single component we can introduce the platform for

the whole repository. In short, all divisions are different and we have to adapt to them (1b).

K: Do you use anything from the JCP-platform that SEP use, can you collaborate with their ways of metrics?
TLA: SEP has focused around the JCP platform. We can collaborate with them with other products; our goal is to have the code for all the java-projects and to have less detail and only the macro-indicators for the code-quality. They want to have a deep focus in detail on their products and we want the general overall picture of the organization. Should we allow this code in production, how should we fix a problem etc.? We need to support the organization (2). We can contribute with more knowledge and data to other division, SEP already knows more than us so they do not need our support (4).

K: Regarding that this program is fairly new; can you see any organizational improvement? Even though it only has been in place for about 6 months?
TLA: Yes, something that has changed is that we now can put a figure on e.g. the number of LOCs in Amadeus (2). Then when management talks they know the figure. We can quantify. Now I start to see the management starting to notice improvement-points from the metrics. Also, they identify the relations between e.g. the lines of code between Java and C and they compare that with the figure they had in mind (3). Then they might ask why, we show them and they can say in which direction he/she wants it to go and we have to monitor so we are heading in the right direction (4).

K: So for now it, at least, raise awareness for the management?
TLA: At least it raises concern in the managements, to see if it confirms with their current view (3).

K: So basically speed up things, if they notice thing they would not have noticed in a couple of months otherwise.
TLA: Yes it speeds some decision processes. Also I think that we should put in place some benchmarking towards an industrial reference (1c, 3).

K: Isn't that hard because of the lack and difference in data that are collected in the industry?
TLA: All these figures have to be filtered and put in relation to their value. But for instance if the industry are producing 1000LOC/month and we 100LOC/month we can ask why and see if it is justified or not (3).

K: I know that it was a report from SPR (Software Productivity Research) within SEP how their results was compared to the industry overall. Do you know it that only was in SEP?
TLA: I think it was one more department but it was not company-wide.

K: Was the reason that they were the most mature division?

TLA: Maybe they are more developers or have a bigger demand in proof of concept with external companies. But I don't know really.

K: To go back a bit, how do you display you collected metrics to the management? Do you send out reports, do they have to ask you etc.?
TLA: It is different levels of reporting for different management groups. E.g. to top-management it is only push, they never pull. We provide them with information, through graphs; it is ad-hoc reporting to top-management. It is highly unlikely that a tool can provide them with exactly the data they want. It is push so we establish a period, and we are validating and pushing the data (2). In the future only push since we assume that validation will be automatic. Then we will open some access to the database and give people standardized SQL request that they can use with some dashboard functionality (Note: Exactly like MS Reporting Services) and the third level is open access to the database with the schemas and you can do whatever request you want. Cause we can't plan all the different request-needs (1c).

K: Is the DB open for now or is that a future step?
TLA: It is open for request. Our goal is to be transparent with the metrics (1c).

K: To go back to the start of this conversation, when you started with this metric-program, what was your first goals, did you start trying to define metrics or ways to collect data?
TLA: It was to answer a simple question from the top-management in Sophia (the top management on Amadeus site in Nice). He asked how many LOC we did have. It is a very basic question like the number of employees in the company. We need to know how many LOC we have and how many are legacy, by domain, etc. Step by step (1a).

K: So it was basically an ad-hoc project - to answer one question.
TLA: It is coming from one question then it was up to us to structure the answer. You could not just give a figure (1a).

K: Okay, then it just grew?
TLA: Little of both, we wanted to answer a question and we went a bit further and then by continuous feedback from the management, when they did see something interesting and new metrics. So it was a kind of iterative process, questions and answers that pushed for something more (1ab, 2).

K: Ah, is that the main reason that you don't have any longer plans than September, because you need feedback from the implementations you do?
TLA: The metrics we have now are clear and we can get feedback on those (1a), but with the quality platform we can't detail that a lot. Cause the divisions are not quite there and we have to proceed and see what is the added value we can give and how we should the proceed with the implementation. As for the central systems we discussed earlier. So once again it will be an iterative process, we do something, get feedback and we continue that way. We don't want to go too far if it would be the wrong direction (1c).

K: Is the proposal solely new metrics or it is also to make the process more efficient?
TLA: Both, we propose new metrics and in the same time we exchange difficulties with them to get the metrics stable for the long term (2, 1b).

K: Do you think you have an established an effective way of collecting metrics today, or do you want something more from your current processes to make them more efficient?
TLA: There are still manual steps in the process that we have to automate, also to communicate the importance of providing us with good meta-data (1b, 4).

K: That everyone pulls in the same direction and respects the importance …
TLA: Yes to know the importance and respecting the delays for us to have them ready. In fact I hope this will be easier now since the top management now gives it higher priority (4).

K: Ok so metrics have a high priority within top management?
TLA: Hopefully they will now have higher since now they know what we are producing. They see added value and gives us okay to continue (3, 4).

K: So you have a budget that you can use?
TLA: We are a team that is dedicated to this so we are not that demanding. But since we are in the central everything for us are “on top of things” (Note: If money is expected to generate more value somewhere else, it will go there), the key for continuing with this division is to provide results so they can see that this is something that help them (3, 4). I think that we have reached that level, since now if the directors want to see this, they know that we cannot provide them with that without their contribution (2).

K: Basically, to be more efficient you need support from top management and get top management to push those that you (the central metric team) are dependent on to provide you with the right data?
TLA: Yes, and we profit when we present the results. But not only the results, we are explaining the process of collecting. We explain that it is sometimes problem with collecting the data (1b, 2).

K: Are all your current processes and tools for collecting data open source, or something developed in-house?
TLA: We have several tools, for counting we use a commercial tool, but it is only priced €100 so it is almost free. It is called RSM. For counting the details, added, modified and deleted LOC we use UCC and for quality platform we use Sonar.

K: I know that Sonar is used within DEV-SEP, is it used in more divisions?
TLA: For now Sonar is only used within DEV-SEP.

K: To go back, you said it is important to get the right data points to be efficient and actually display relevant data. Could you say that efficiency is divided by two things,

the organizational support and the tools you are using? Which one of those two do you think is the most important part of efficient metric collection?
TLA: It is both but if I had to choose I would chose the support of the organization. If you have the support and resources from the organization you can always find the tool (4).

K: The organizational support is the biggest obstacle for all companies …
TLA: I think it is, to have the right priority and this is something they want to do, all the quality that is, but it hard cause of other business priorities. It is hard to put things in the middle of business things. You have to be convincing, show the benefits, you have to support them and you have to invest yourself by doing some development. That is key (4). Tooling is required and if you would need to develop a new tool it would be a big investment but we see today a lot of things happen in open source. However, it could be an expensive tool that was a big investment, and we are a company whose main goal is to get customers. We are still in our expansion phase since we are still increasing on all areas. So the quality is necessary but investing a lot … Let's say you have 15 million, where should I put them, to make an action plan for the metrics or try to get new customers. For now with new markets it is difficult to justify big investments in quality. But if you wouldn't be in an expansion phase you could dedicate resources to internal projects. The money will go where it can create the most value, and new customers, development and new markets is, at least for now, more lucrative than increased monitoring of the internal quality.

TLA: If our quality is bad we have to do something about it, but if it is something that will generate a return that is very long term, it is hard to invest a lot of money (x).

K: Even though you are still expanding as a company, could you say that it is an organizational push regarding metrics and quality since you had this iterative process?
TLA: The initial push is to have the visibility, and then it is a second decision to what we are going to invest to improving the quality. First they see and if the situation is not very critical they compare with other thing they could do with that money. But they push for the initial status, were we at. They, the top management, need to know (2). That's why the quality platform, our goal for 2011 is to show the figures regarding quality so we know that, at that time they will look and decide to start with the quality platform or not. If we start the quality platform and gets good result it will imply investment, in money, resources, people, etc. They invest if they will get a good return of their money; maybe spend less on maintenance etc. But, as said, this is only the second decision since they constantly compare the ROI (return of investment) (1c, 2).

K: Ok, so the current metric-process has been small organizational pulls? Like can you do this, can you do this, etc. then you deliver this quality statistics and then it is a big pull from the organization.
TLA: It is an iterative process; you answer the question and give them a little bit more. Push and pull relationship. The quality platform is a milestone were we have to provide the indicators (orange, red, yellow) if things are yellow they will probably

not invest that much, It is important for management to see the status, the decision is not only regarding quality but it is much larger (1c, 3, 4).

K: Do you have an ultimate goal for the metric process? E.g. we would like to be able to monitor x and y or is it a continuous process that is becoming better every year.
TLA: We want to put a process in place to make sure that we are improving (1c, 2).

K: What are you most happy about concerning your current way of collecting metrics, or with the metrics program in general?
TLA: When I see top management surprised by the figures.

K: Ok, then the other way around. What are you the least satisfied with?
TLA: The lack of automation in certain domains. That is not directly connected to the metrics. On certain areas the development process is not that precise so when we ask for the meta-data it is really painful since some people are not managing a coding repository and the directory is not as precise as it could be (1b).

K: So the coding-process is lacking definition?
TLA: Definition or ensure that all of the meta-data is in the central repository.

K: So they do not use ClearCase (Note: ClearCase is a code repository system used within Amadeus). I imagine it would be easier if all used the same system.
TLA: It would be easier but it is not a driver, since we will never get everyone to use the same tools. We know that we have to adapt. When we ask for the meta-data, we are suffering from the divisions internal priorities. They are not always ready to provide us with the input that we would need. But hey, that's life in a company. We can't forget that these guys mission is to have the division up and running, this is a secondary goal. They have to provide us with data but the production will always be priority one. Our questions are important but we have to be flexible and adapt (1b, 4).

Transcript Group B

Date: 22/3 - 2011:
Codes:

5. *Process – concern the way that they are working with metrics*
    a. *Past*
    b. *Current*
    c. *Future*
6. *Product – concerns the actual results from the metric work. The end product that is delivered.*
7. *Performance – How are these metrics helping the organization today and how are the used, respected or not.*
8. *Key areas – areas that are critical for the continued work with the metric program.*

K: Kristian Mattsson
TMB: Team Member Group B
TLB: Team Leader Group B

About System Wide Adaptive Testing[4] (SWAT):
TMB: The main purpose is testing, the idea is to parse log-files and put them into a database.
K: For how long has your SWAT-project been going?
TMB: Less than 5 years.

K: Is it designed for collecting software metrics?
TMB: There are many purposes, first to identify something that went wrong. E.g. a PNR, it is a tool that lets us collect everything about a PNR. It is not statistic in itself.

K: Is the reason behind SWAT that you did not have any way of monitoring your data-process, and you got software metrics from SWAT?
TMB: Not software metrics SWAT it mainly for collecting data for testing purposes and is able to replicate a test-scenario. The idea is that you collect log-data, you analyze that data and you automatically rebuild tests. It is called model-based testing approach and is very complex. From this the idea is to use this data to get some statistics.

K: So the data that you are collecting is only for this test-purpose, or is it other purposes as well?
TMB: (Regarding SWAT) initially it was only for testing, for now it is also used for doing some BI around that stuff. The idea around BI is to be able to understand the amount of traffic between all Amadeus-servlets. It is not for testing developers, only for Amadeus products. Testing developers and machines is a totally different subject.
(One in the department) is working on this subject and we try to sell it to other teams within Amadeus, which it not always easy. And there are many other initiatives regarding BI within Amadeus. All the different teams are saying that they are doing BI (1b, 2).

K: But your specific BI is for monitoring traffic between the servlets?
TMB: Yes.

K: But as I understand you are also using Software Management Tool[5] (SMT)for statistic?
TMB: SMT is completely different statistics. (1b)

K: Is SMT another product that you are selling to other teams?
TMB: Yes, the purpose of this team is to drive the builds, tests, and everything else around source-control. Everything around support for the developers and that is some statistics. Are the tools provided to developers a good ROI, are they used, are they good (1b, 2)?

K: Was SMT developed with the purpose of selling to the other departments?
TMB: SMT was developed to help developers with their day-to-day activities. With SMT we have a way to see if the software is used or not (1a). The purpose is more to, since we collect several things, know how long time it takes to compile/install/etc. And there are a lot of statistics that can be derived from this (2).

K: Do you use that statistics today for something?

[4] SWAT is a testing initiative and was not developed for software metrics but rather collect data to recreate development scenarios that generated errors. SWAT is not in the scope of this study and is primarily included to provide more information about the context.

[5] SMT is referred to as ”Devtool” in the report.

TMB: We use the statistic for monitoring the acceptance of the tool, to see if it is accepted or not, if it is used or not? (1b, 2, 3)

(TMB shows graphs generated by MS reporting services from the SMT-database)

TMB: It is for monitoring if it is worth investing in this kind of tool (3).

K: So a general experiment, like a proof of concept?
TMB: Like return of investment. Do we invest in this tool? Does this tool help developers or not? (3)

K: When did you start developing this tool?
TMB: It was originally developed by someone in [another division within the company] and we took the leadership of this tool June or July 2010. We have worked on it since July (1b).
TLB: In the beginning it was only used by (another team in Amadeus) and we have now produced it for several teams within Amadeus (1b).

K: Could you say that it has generated a good ROI this far?
TLB: This far yes (3).
TMB: This far yes but I would like to increase the number of users of it, today there is maybe 120 developers using it of a total of 400-500 (rough estimate) (1c). We also have statistics about build time, etc., all data that comes from using Jenkins/Hudson[6]. The data we get from Jenkins is free for us, it collects all the data. With SMT it is different since we developed it we collect data and generate reports (2).

K: But for now you only use statistics regarding the number of developers using it?
TMB: Yes, even if we collect a lot of things we only use that data (1b, 2).

(Talk about how they decided to invest in a tool. Group B pushed for extra statistics. However, the BI (Business Intelligence) on top of that is not a priority from the division (4).)

TMB: Two aspects with BI today, draw the graph of the data-flow, click-stream, from the top servlet to the end (1b, 2, 3).

K: That was SWAT?
TMB: No, SWAT is the source of the information. SWAT contains a set of log-file parsers. SWAT populates databases from which we design a data warehouse database which serves as a basis for drawing graphs during different kinds of information, extrapolating the data.
TLB: We have different topics to deliver and improve the build-time and quality of the development environment and benchmarking some workstations, monitor several models of workstations (3). With SMT we are normally able generate a graph per workstation, product and release (2). I have generated a couple of graphs for this to be able to come to the top management in the future and …

K: Have you shown these graphs (regarding build time for different workstations) for the top management?

[6] Jenkins/Hudson is a continuous integration tool which support SCM tools including CVS, Subversion, Git and Clearcase and it can execute Apache Ant and Apache Maven based projects. See http://hudson-ci.org/ for further details.

TLB: It was an action from my side to say that we need to replace workstations more often, and the reason behind that action is the question “are we providing sufficient hardware to people?” A new investment depends on if the action that takes time is done/taken by a lot of users or not (3, 4).
TMB: For me there are two different directions.

K: With SMT?
TMB: With SMT and build-times.

K: So build-time is your primary statistic?
TMB: The first one is with SMT, to monitor a set of well controlled machines, the model, the release …(1b, 2)
TLB: If I have a graph generated every week on the build time for a reference machine, then we can directly detect if there is trouble with deliveries etc. (2, 3).
TMB: Also to see the evolvement of maven or weblogic (the web-server that is mostly used within the organization). What will be the gain with maven 2 vs. maven 3 etc. The first step is to benchmark machines to see the evolution; we are more invested in this alternative then the second one I am going to tell you about now. Which is a wish from (manager in the org), and he is not interested in benchmark data, he want to know what is really happening on real developer machines. SMT could help with this problem; it should be used more in this sense of what is happening on developer machines (1b). By doing a couple of graphs around SMT-statistics you could find out why things happen.

K: Where do you want to go with SMT, like in two years? Let’s say that you will get an investment from the management.
TMB: I think SMT is developed as times goes by, it should be redesigned on day and we are working on to make the tools for the developers better and better. If the developers do not like this tool, we will not use it anymore (1c). There is a lot of complex task, some of SMT should help with, and we have to find a way to show them that they can save time with SMT (3,4).

K: So the statistics of SMT is a secondary thing?
TMB: Yes. We have to study the analysis axis, there are a lot of actions, times and specific configurations and you have a lot statistic (2).

K: Yes. But you said that if developers may not like it, they don’t have to use it. Can’t the statistic part be an incentive to some extent push SMT to the developers?
TMB: That could be the case but so far that is not the case. We always try to enhance SMT so it corresponds with user needs. Are developers using it; we have to understand why so we can try to improve it. And if we if we improve it, will the developers go back to using it (1b)? We can’t impose anything and have to seduce them from the potential benefits (4).
TLB: You need to understand that as a service team, with few employees that services a big part of the organization, we can invest in one tool. Not ten. If other divisions want to invest in something else, they can (4). But that would probably be more costly to Amadeus in the end. We need to adapt and therefore only support one tool that makes that possible for us. If we change a tool it only need to be done once since the other divisions can take part of it. We need to keep in mind that having a tool developed by all teams will cost more in total then having a tool located in one place (1b).
TMB: That is right but you have to keep in mind that the objective for the other departments is to deliver their product in time. As long as they do that it’s fine. It is up to us to show them

the gain by using SMT and we have to put in place statistics around this and show the gain (2, 4).

K: So for now the statistic that SMT collects it to show that it is used?
TLB: That is the only report we have.
TMB: But there are plenty of nice reports that you could exploit.
TMB: To collect the information for the compiling time for each maven-module, we either use SMT as I told you or we can instrument maven. Not all people are using SMT but all are using maven (1b).

K: Wouldn't the most efficient way be if everyone was using SMT?
TMB: Yes, that is the point. If you manage and provide a tool that developers has to use, and if it collects statistics, you have to be very careful so that the statistics-collection do not slow down the execution (2, 4). That is what annoys me about instrument maven.
TLB: If we do that we have to recheck and rework when someone switches to from maven 2 to maven 3.
TMB: In addition, if someone wants an older version we suddenly can't collect statistics anymore, but we could if they use a tool that we control and the end-user is using.

K: It really sounds like the SMT tool should be the best choice.
TMB: Personally I want to focus on SMT and show management that SMT collects statistics and if we want statistics we should use SMT. It is calling itself, management should push SMT only to get the statistics (2, 4).

K: Do you know if they know that?
TMB: This far statistics is not known outside of this team or when RK presents the statistics for the management (1b). Only you and [another employee, the one who developed SMT in the first place] know about the statistic.
TLB: Also some people from the DEE.P committee (an internal organ for projects and initiatives regarding the developing environment), which is a bit strange since they want to collect the data but reporting is not a top priority for the moment. That is strange since they want the data but we never export the data (2, 4).

K: Ok, let's say that my manager would ask you for data or a report, could you provide it to him?
TLB: I could provide him with a PP-presentation but he would not get access to the database due to legal aspects. It is a matter of weeks of development and tie work processes to that. We need tutoring on the processes to quicker be able to detect when something happen. We need time and money to do that. It would be easy to spot degradation in the workstations but it is a lot harder to improve the performance (1b, 2, 3, 4). There are a lot of alternatives …
TMB: You have to keep in mind that this team is for providing support for developers, this team is naturally a team that collects data and we should provide reports to management. Too monitor the effectiveness of the teams (1b, 2).

K: Is it a lot of ad-hoc statistics project in this company in general?
TMB: You could say that every division is closed (towards the others) down and is its own sub-company. In SEP which shares at common platform we have a couple of projects (4?).

TLB: we are tied to political games. We have all the data, for example in Sonar that we could generate BI-reports from, we have a lot of data but no reports to extract (1b, 2, 4).

K: Is it no one that asks for the data?
(Showing of some of all the data-sources and statistics they have)

K: You have data but nothing to do with the data? What I understand you collect huge amounts of data through Sonar, SMT, etc. but there seems not to be any action to take with the data.
TLB: We have the data, having the data is one thing and having the report is another, better, thing. But having actions associated to a report is really what we have to target (3, 4).
TMB: To do the actions you need to ask the questions, what do I want to see? (4)

K: So your goal is to get from report to action?
TMB: No we have some reports but the idea would be more of drawing reports with respect to the data that we already collect.
(Sidetracked by more graphs, data is exploited and could generate a business report for corrective actions.)
TMB: SMT is not for business reports, it is for internal development and issues regarding that. It is another subject, different area (2, 3).

K: So your reports are basically just used internally to maintain the development environment?
TMB: [Other members in Group B] is working on reports for top management.
(Showcasing reports which are pushed to top management. They contain build-times, compilation errors, etc.)
TLB: Top management wants to know that the quality of service is good. For development it is not the same.