Combining Intranet Data Warehousing and a Traditional Approach in Metrics' Implementation: A Case Study from Nokia

Arto Oinas
Nokia / Fixed Switching, Elektroniikkatie 3, 90570 Oulu, Finland
arto.oinas@nokia.com

Abstract

The focus of the software measurement research has been in the definition, choice and data collection of metrics. So far, less attention has been put on improving the reporting and analysis of metrics. For that purpose, intranet data warehousing is a promising new technique. Fixed Switching unit from Nokia started a program to develop fault metrics, and in the program, intranet data warehousing was used to implement some metrics. Besides to that approach, an operations system query tool and a spreadsheet were used to implement other metrics. Intranet data warehousing was found useful in implementing metrics that contain trend data, have a large user base and have short reporting frequency. The users can access the intranet metrics with standard web browser, and only little training is needed. The analysis of metrics is supported by dynamic on-line reports. Using the query tool and the spreadsheet is a cost-effective although modest way to start metrics’ implementation. For the measurement needs of a software developing unit, intranet data warehousing is expensive solution, and it cannot be considered before there is a reliable corporate data warehouse. The development should be started from where the benefits are the most visible.

Keywords: data warehousing, software measurement, intranet

1. Introduction

Measurement is a common approach to software process improvement. According to the Capers Jones' [8] survey of software industry in the United States, measurement is among the best means to improve productivity, software quality and schedule accuracy. There are many success stories on applying measurement; e.g., corporate-level programs at Hewlett-Packard [4, 5] and Motorola [3] have resulted in significant improvements in quality.

Experiences [5, 13] suggest automating data collection for metrics whenever possible. Large software developing organizations have operations systems supporting their main processes. Fault tracking and project management systems are examples of these systems. Although the systems can be used to collect metrics’ data, they have not been designed for that purpose, whereas a data warehouse is intended specifically for data collection in support of management’s decisions [6]. In a data warehouse, the data is organized by subject instead of specific business process as in operations systems. Data warehouse integrates both the data content and definitions from multiple operations systems. Operations systems by themselves rarely support integration because the systems are often based on different technologies. The data warehouse data is non-volatile that means that, once data is loaded, it is kept for a long period. In the typical operations system, data is often kept for a short period because it is important only at that moment. In the data warehouse, it is important that the data is time-variant so that all data can be related to a specific time point or interval. Operational data does not always have an explicit temporal dimension. It might not be interesting for an operations system to know when a transaction took place or when it changed its state.

Leading organizations use tailor made tools to produce metrics [11]. These tools often accompany operations systems and allow making simple reports and graphs. Data warehouses support using the normal reporting tools, but in addition, they can be used to support on-line
analytical processing (OLAP), that is, analyzing corporate data intuitively in multiple dimensions [2, 15]. OLAP has previously been the privilege of data warehouse experts, but intranet data warehousing promises to be a solution how a large number of users can cost-effectively and easily do on-line analytical processing [2, 15]. An intranet data warehouse is a combination of techniques that enable users to dynamically generate a database query, conduct data analysis, and format the results as text or image files for display in any web browser [15]. There are some experiences in the technical implementation [7, 10] and in the practical applications in enterprises [15].

According to Pfleeger et. al. [13], the focus of measurement research has been in the definition, choice and data collection of metrics. So far, less attention has been put on improving the reporting and analysis of metrics. Therefore, researchers are encouraged to fashion results into tools and techniques that practitioners can easily understand and apply. Intranet data warehousing could be this kind of tool. However, a new technique like the intranet data warehousing can have its problems that are not only technical ones like intranet security and the scalability of hardware and software [15].

This paper describes the experiences in implementing metrics within the fault metrics program of Fixed Switching unit from Nokia. There are two implementation approaches. The first approach is intranet data warehousing, and the second approach is based on using a database query tool and a spreadsheet. First, the technical implementations are presented. Then, the benefits and issues of using the approaches in the data collection, reporting and analysis of metrics are compared. Finally, the development costs of the approaches are compared, and issues in conducting an implementation project are considered.

2. Case Presentation

The research and development unit of Fixed Switching (FSG) develops software for DX 200 fixed network switch, which is a large real-time system that has been developed since the early 80’s. The system contains over 14 million lines of code. FSG has launched several metrics development programs to start measuring its main processes. The fault metrics program is one of these programs. Earlier in the improvement program, the metrics have been defined [12], and now it is time to implement the first metrics.

FSG has a long history in using operations systems that are specially tailored to support its processes. These systems include the fault tracking system, version control system, project management system, and the delivery database. A corporate data warehouse integrates data from the operations systems used by several R&D units. Considering the systems used in FSG, only the fault tracking system has been integrated to the data warehouse. In this case, the fault metrics are based on the data of that system.

The fault tracking system, PRONTO, is used throughout the fault management process in FSG. In fault management, internally found and customer found software problems are recorded in the system as problem reports, and assigned to the group that is responsible for solving them. Each problem report has a state, which is updated as the group investigates, solves and corrects the problem report. When the problem is solved, the solution is recorded in a problem analysis. It includes the cause of the failure and a response that is sent to customer. Finally, when the problem has been corrected, a change object containing the name and version of the corrected program block.
3. Metrics' Implementation with Intranet Data Warehousing (Approach 1)

This section presents the technical solution used to implement metrics with intranet data warehousing in the case of FSG. The section covers the data warehousing architecture, the intranet deployment architecture, and the web user interface with which the users can access the metrics.

Figure 1 depicts the data warehouse structure. In the data warehouse, the problem report data is organized in multiple levels of detail. Operative data warehouse has the highest level of detail. The latest problem report data flows from the fault tracking system into the operative data warehouse daily. Older operative data is stored in a table containing changed problem reports. Unfortunately, that table does not capture all needed change data, which is fragmented in several tables. The size of the operative data tables is huge, and it is not practical to use the operative data directly for the fault metrics of FSG. Therefore, the data needed for the metrics is loaded into a second level of detail. This kind of specific data is called the datamart level [6]. Above the datamart level, there are two lower levels of detail. There are detail data and summary data tables for each metrics report. The detail tables contain problem report level data. The summary tables aggregate the detail tables. Their values correspond to columns in the metrics’ reports. The detail and summary tables contain also organizational levels of detail. The levels are implemented as separate detail and summary tables for the unit, department and group level data.

Figure 2 presents the intranet architecture used to deploy the metrics. The architecture is partitioned into three tiers. Data warehouse is the data tier, the data warehouse server provides the logic tier, and the web server and the web browser form the presentation tier. The user accesses the application with the web browser that submits a query in the form of an HTML-encoded request to a Web server. The server in turn transforms the request into a Common Gateway Interface (CGI) script, and sends it to the data warehouse server. An user ID and a password are asked for that
access. Data warehouse server executes the query script on the data warehouse data, and retrieves the results. Depending what the user has requested, the data warehouse server returns a pre-calculated static report or it creates dynamically a report. The web server receives the report, combines it with other HTML content, and sends the page to the browser. The data warehouse server creates static reports periodically and stores them.

Figure 2. The Intranet Deployment Architecture of Metrics in the Case of FSG.

Figure 3 shows the user interface of a dynamic metric report in the web browser. The user interface consists of the control frame in the left and the report frame on the right. In the control frame, the user sets wanted values in the fields and requests a dynamic report. The report frame shows then results of the query. When not showing the results, the right frame shows information about the metric including usage instructions and a link to the definition of the metric.

Figure 3. User Interface Showing Metric Department Fault Traffic.

The metrics in Figure 3 is department fault traffic. It displays the number of incoming, handled and open problem reports of a department in the last thirteen months. There are several possibilities to explore the metric. First, it is possible to select whether to show a general view of the fault traffic, a detailed view of the incoming problem reports or a detailed view of the handled problem reports. For example, the detail level of the incoming problem reports consists of the new problem reports and the problem reports directed from other responsible group. Second, the organization whose problem reports are examined can be chosen, and the user can adjust the
organizational level from department to an individual group. Third, it is possible to restrict the metric to a certain software release, to internal problem reports or to customer problem reports. Finally, as the selections have been made, it is possible to either make a graphical report or to a drill-down into the columns of the graph. The latter means breaking the monthly columns into individual problem reports and showing them in tabular format.

4. Metrics’ Implementation with a Query tool and Spreadsheet (Approach 2)

This section presents an approach (Fig. 4) which uses a query tool, PRONTO Reports, and an Excel spreadsheet to produce metrics. This approach to produce metrics is used in the quality reporting of FSG. Quarterly, the quality personnel in the organization use the query tool to extract data from the fault tracking system PRONTO, and the spreadsheet to create charts based on the data. Then, the personnel copy the charts to their quality report, and present the results to the organization in question. The tools used the approach are already available to the practitioners.

The fault tracking system and the query tool are Lotus Notes client/server applications. The user accesses the applications, which reside on network servers, with a PC. The query tool allows defining searches on the data of the fault tracking system. It is also possible to define which fields are included in the output and what output type is used. Output type can be either readable ASCII or tabulator delimited data that the spreadsheet can read. Normally the query tool is used for ad-hoc reporting but it in this case that is used to produce data for metrics. For that purpose, the person producing the metric needs detailed work instructions how to adjust the query and how to export the results to the spreadsheet.

![Figure 4. Using The Database Query Tool and The Spreadsheet.](image)

Figure 4. Using The Database Query Tool and The Spreadsheet.

The fault tracking system and the query tool are Lotus Notes client/server applications. The user accesses the applications, which reside on network servers, with a PC. The query tool allows defining searches on the data of the fault tracking system. It is also possible to define which fields are included in the output and what output type is used. Output type can be either readable ASCII or tabulator delimited data that the spreadsheet can read. Normally the query tool is used for ad-hoc reporting but it in this case that is used to produce data for metrics. For that purpose, the person producing the metric needs detailed work instructions how to adjust the query and how to export the results to the spreadsheet.

![Figure 5. Spreadsheet for the Metric False Problem Reports.](image)

Figure 5. Spreadsheet for the Metric False Problem Reports.
Figure 5 shows one spreadsheet workbook used to produce charts from the queried data. There is a workbook for each metric. The query result data is imported by launching a macro with the button Import Data. The macro reads the data and automatically re-calculates data tables used in the metrics’ charts. Then the user can select wanted chart, add title information and copy the chart to a document.

Figure 6 shows one metric, false problem reports, produced with the approach presented in this section. False problem reports is the percentage of problem reports, which have not caused a corrective action, to all handled problem reports. The trend of four quarters is shown. In general, it is not possible to get trend data from the fault tracking system because it contains only current data. However, problem reports in the system include a field capturing the date of last state change, and in this case, that field can be used to form time-series.

![False Problem Reports Graph](image)

Table 1 compares data collection and storage in data warehouse (Approach 1) and the fault tracking system (Approach 2). Data warehouse makes it possible to implement metrics that use long-term trend data. However, data warehousing being yet an immature activity in the company, there are some disadvantages. The fault tracking system can be used as a data source if metrics need only the current data, or require trend information restrictedly.

5. Benefits and Issues

Two approaches to implement metrics have been presented, intranet data warehousing (Approach 1), and using the query tool and spreadsheet (Approach 2). Firstly, this section compares the benefits and issues of these approaches in the data collection, reporting and analysis of metrics. These phases are refined into data collection and storage, producing and distributing reports, and presenting and analyzing the results [1, 9, 14]. Secondly, this section compares the development costs of the approaches, and considers issues in conducting a metrics’ implementation project.

Table 1 compares data collection and storage in data warehouse (Approach 1) and the fault tracking system (Approach 2). Data warehouse makes it possible to implement metrics that use long-term trend data. However, data warehousing being yet an immature activity in the company, there are some disadvantages. The fault tracking system can be used as a data source if metrics need only the current data, or require trend information restrictedly.
Table 1. Comparing the approaches in data collection and storage

<table>
<thead>
<tr>
<th>Approach</th>
<th>Benefits</th>
<th>Issues</th>
</tr>
</thead>
</table>
| (1) Data Warehouse | + Provides long-term data by integrating data from the old and the new fault tracking systems  
+ Once loaded data warehouse data does not change  
+ Good in trend metrics because provides historical data about problem report state changes and responsible group changes | – Misses rapid changes in data, e.g., urgently handled problem reports, a snapshot of data is taken once a day  
– There have been problems in loading data into the DW, and that causes false peaks in the corrected problem reports  
– Due the ad-hoc development of the operational data warehouse, its architecture is complex |
| (2) Fault tracking system | + Provides current state data of problem reports, provides also some time stamped data including creation dates and last state change date | – Does not capture problem report state or responsibility changes  
– Using data for metrics is restricted to the data that is not likely to changed anymore  
– No long term data |

Table 2 compares the advantages and disadvantages of the two approaches in producing and distributing reports. Intranet data warehousing (Approach 1) gives benefits of scale, minimum manual effort is need to produce and distribute metrics for a large number of users. Metrics can have very short reporting frequency. Users are familiar with using web, and little training is needed. Dynamic reports are useful for delivering metrics that have several organizational or other points of view. Using query tool and spreadsheet (Approach 2) is recommended if the reporting frequency of the metric is relaxed and there is a delivery channel like the FSG quality reporting. Besides, quality report provides a permanent record of results.

Table 2. Comparing the approaches in producing and distributing reports.

<table>
<thead>
<tr>
<th>Approach</th>
<th>Benefits</th>
<th>Issues</th>
</tr>
</thead>
</table>
| (1) Intranet data warehousing | + Minimum effort to distribute metrics to a large number of users.  
+ No training is required. The users are familiar with the web user interface.  
+ Enables short reporting frequency.  
+ The dynamic intranet reports allow having metrics that have many views, e.g. for the different organizational perspectives. | – No disciplined storage of old reports.  
– Intranet security is vulnerable; however, there is access control.  
– Performance slows as the number of users increases. |
| (2) Query tool and spreadsheet | + Only standard office applications are used.  
+ Quality reports are stored in database that has version control.  
+ Once created, quality reports can be distributed by providing a Notes-link. | – Requires manual work.  
– Practitioners require more training in using the tools and in producing the metrics. |

Table 3 compares the two approaches in presenting and analyzing the results. The most important benefit of intranet data warehousing (Approach 1) is the possibility to do on-line analysis of metrics. However, the outlook of intranet reports is modest. On the other hand (Approach 2), the spreadsheet provides better facilities for drawing reports. For ad-hoc analysis, query tool and the spreadsheet are recommended.
Table 3. Comparing the approaches in presenting and analyzing the results.

<table>
<thead>
<tr>
<th>Approach</th>
<th>Benefits</th>
<th>Issues</th>
</tr>
</thead>
<tbody>
<tr>
<td>(1) Intranet Data Warehouse</td>
<td>+ On-line analysis of data by changing the point of view (organization, release, customer)</td>
<td>– Response times in drill-down into table data can take few seconds, and this can frustrate the user.</td>
</tr>
<tr>
<td></td>
<td>+ Multidimensional architecture allows drill-down into data</td>
<td>– Outlook of the bitmap format reports in web page cannot be easily changed</td>
</tr>
<tr>
<td></td>
<td>+ The reports and the analysis instructions are available from the context.</td>
<td></td>
</tr>
<tr>
<td>(2) Query tool and spreadsheet</td>
<td>+ Query tool is good in ad-hoc analysis of current fault data</td>
<td>– Ready metrics are rather static, on-line analysis is not possible</td>
</tr>
<tr>
<td></td>
<td>+ Spreadsheet offers rich features for data analysis</td>
<td></td>
</tr>
</tbody>
</table>

Now that the benefits and issues of using the approaches have been compared, it is worthwhile to compare the development costs of the approaches. In addition, guidelines are formulated for conducting metrics’ implementation projects with the approaches.

- Using a spreadsheet and a query tool accompanying an operations system is a cost-effective way to start implementing department and group level metrics. Spreadsheet allows prototyping metrics [11] based on user feedback and changes in the business environment. If a metric is found useful, it can be later automated.
- Before implementing metrics with data warehousing can be considered, there has to be a corporate data warehouse providing reliable data from the most important operations systems. An earlier metrics' project exceeded its schedule and budget by more than 100% because of the bad quality of data warehouse data, which could not be used in the metrics that had a high reliability criteria.
- In a corporate data warehousing effort, the know-how concentrates in the data warehouse developers. Outside them, the awareness of the data warehousing is low. Intranet deployment brings the advantages of data warehousing to the grand public.
- The resources of the data warehouse developers go to the general development. FSG hired subcontractors to implement the intranet metrics. Because of needed expertise, the costs become relatively high.
- It is useful to start intranet data warehousing small and from where the benefits are the biggest and most visible in the organization. For example, implementing the metric department fault trend automates a metric that has been previously produced manually, and has a large number of users.
- Corporate data warehousing gives economies in implementing executive metrics collected from several R&D units. Considering R&D unit-level metrics, it is beneficial if the same implementation is adaptable for the needs of multiple units.
6. Conclusions

In the FSG fault metrics program, two approaches are used to implement metrics. The first approach is intranet data warehousing, and the second is using a query tool to extract data from the fault tracking system and a spreadsheet to draw metrics. Intranet data warehousing offers benefits compared to the other approach in the data collection, reporting and analysis of metrics. Data warehouse can collect long-term trend data that is not obtainable from the fault tracking system. However, if data warehousing is an immature activity in the company, there can be problems with the quality of the data. In the reporting, intranet deployment gives benefits of scale. Minimum training and manual effort is needed to produce and distribute the metrics for a large number of users. Dynamic reports allow doing on-line analysis, although that requires hardware processing capability. The second approach, using the query tool and spreadsheet, can be used if the metrics require historical data only restrictedly, and the reporting frequency is relaxed. Besides, the spreadsheet provides better facilities for drawing reports than intranet, and for the ad-hoc analysis of the current data, the query tool and the spreadsheet are recommended.

Implementing metrics with intranet data warehousing is expensive, and before it can be even considered, there has to be a corporate data warehouse providing reliable data from the most important operations systems. Compared to that, using a spreadsheet and a query tool accompanying an operations system is a cost-effective way to start implementing metrics. With that approach, metrics can be prototyped and if a metric is found useful, it can be later automated. The development of intranet data warehousing should be started small and from where the benefits of automation are the biggest. In corporate data warehousing, it is beneficial if metrics can be implemented so that they are adaptable for the needs of multiple software developing units.

This paper has covered only the implementation of the first metrics within the fault metric program of FSG. So far, there are no experiences in using the metrics, and it is not known what are actual benefits of these metrics to the organization. Considering the technical aspects, there are only experiences in integrating data from one system, the fault tracking system, to the data warehouse. In the reporting and analysis of metrics, web technology has yet much more to offer, for example, Java technique could be used to improve the presentation of the metrics and to enhance collaboration.
7. References


