History and Lessons Learnt from a Metrics Program at a CMMI Level 3 Company

Matthias Vianden, Horst Lichter, Simona Jeners
Research Group Software Construction
RWTH Aachen University
Aachen, Germany
{vianden, lichter, jeners}@swc.rwth-aachen.de

Karl-Joachim Neumann
Process Management and IT-Compliance
Generali Deutschland Informatik Services
Aachen Germany
karl-joachim.neumann@generali.de

***Abstract*—Metrics and especially metric-based monitoring dashboards provide valuable information and insights for managers in software development organizations. However, implementing and launching a companywide metrics program is very hard and time consuming. This paper describes the history and our experience with the development of a metrics program at Generali Deutschland Informatik Services, a CMMI level 3 certified company. We also provide important lessons learned alongside a list of consolidated best practices for the implementation and maintenance of a large metrics and dashboard program. We believe that both are useful for every practitioner and researcher in this field and help to build better and more sustainable metrics and dashboard development processes and infrastructures.**

Keywords—metrics; project management; dashboards

I. Introduction

Metrics are an important means to measure the quality of both the development processes and software systems. Improvement reference models such as CMMI require that software development companies build up abilities to systematically apply metrics to support project management as well as other cross cutting software engineering disciplines [1]. Based on quantifiable metrics process managers are able to identify processes that contribute to project success or failure. Hence, metrics are a necessity for objective process improvement. Sadly, it is often difficult to find the *right* metrics and provide *good* measurements. It is, however, even more difficult to steer and control a complex metrics program in a large company over a long period of time. This paper reports on the history and our experiences in establishing a metrics and dashboard program at Generali Deutschland Informatik Services (GDIS) over the last four years.

We organized the papers as follows. In the next chapter background and related work are given. Afterwards we describe the history of the metrics program at GDIS. Based on this we present in chapter 4 the main lessons learned. This leads to a list of best practices which we provide in chapter 5. The last chapter concludes and outlines our future work.

II. Background and Related Work

Over the last 30 years different researchers contributed important work on metrics and metrics programs. Methods like the famous GQM [2] and its modern variations like GAM from Cyra and Górski [3] may help to gather requirements for metrics and dashboards. Unfortunately, they are far away from a complete engineering approach which at least should include requirements engineering, development, testing, operation, and maintenance. Münch and Heidrich successfully and thoroughly worked on metric dashboards [4]. They also proposed a GQM based method for the *setup* of the dashboards [5]. Unfortunately the proposed approach is heavily rooted in GQM and ignores modern software engineering ideas such as incremental and iterative development or prototyping. Also the maintenance and operations phase of dashboard systems is not included in the process.

An important aspect for a successful metrics program is the specification of those metrics which can be used throughout the company. Hence, the topic of metric specification is addressed in a lot of research papers concerning metric documentation [6], [7]. Most of these approaches are based on metric meta-models or on metric ontologies resulting in a more formal specification rather than informal plain text. However, our experience shows that most of the metric specifications used in industry (if they are used at all) are plain text documents. Sometimes these documents are on a more formal level, i.e. containing dedicated sections for specific attributes. Example are the twelve steps to useful software metrics by Linda Westfall [8], the required specifications for CMMI [1], or at least “goal”, “question” and “metric” sections [9].

III. History of Metrics Program at Generali Deutschland Informatik Services

Metrics are an important means to steer, control, and monitor projects. This is reflected by the Capability Maturity Model Integration (CMMI) which requires to establish and define metrics at project level in the first stage (level 2) [1]. However, a lot of practitioners argue that the most important level to reach for a company is level 3. That is because level 3 requires companywide standards for processes and artifacts. This also includes metrics.

Software Engineering Conference (APSEC), 2013 20th Asia-Pacific , vol.2, no., pp.73,78, 2-5 Dec. 2013
doi: 10.1109/APSEC.2013.115

Even though we are reporting on the history of a metric program a lot of work that we did focused on dashboards. Unfortunately, today the terms *dashboard* is used inflationary throughout several applications. To avoid confusion we call the metric based dashboards in this paper *metric-based monitoring dashboards* (or *M² dashboard* for short). This also emphasizes their purpose at GDIS which is to monitor important aspects of a project or the complete software development of the company.

A. 2009 / 2010 - Early Phase and Foundations

The metrics program was always part of the backbone of the initiative to reach CMMI level 3 at GDIS. As part of this initiative the measurement and analysis team at GDIS in 2009 started to improve the documents and processes regarding measurement and analysis. This of course required a thorough analysis of the existing artifacts. Most importantly metric specifications, measurement sources, dashboards, and metric processes were analyzed.

1) Analysis Details

The analysis showed a lot of interesting results and insights into the current status of the metrics and measurement program.

- IBM Rational development and design tools were the mayor source for measurements. Requirements were stored in Rational RequisitPro[1]. Issues and defects are tracked in Rational ClearQuest[2]. Rational Software Architect [3] was the main tool for design and development. Additionally Tricentis TOSCA[4] was used to design and track tests and test results. A lot of additional tracking information was implemented in various Excel spreadsheets.
- The important project management data was aggregated in an Excel spreadsheet, the so called *project data sheet*. It contained data that had to be added manually as well as automatically added data from other sources like spreadsheets containing project risks or csv files from ClearQuest dumps. Most importantly the project data sheet also contained the Earned Value Analysis [10] metrics of the project including budget and cost tracking as well as milestone tracking over time.
- There was no template for a metric-based monitoring dashboard for project management. The project data sheet contained some monitors for the visualization of important metrics but this was far away from a real M² dashboard.
- Most of the metrics were documented in a large Word document - the *GDIS Metric Model.* This document was a valuable resource when we analyzed the metric situation because it often clarified the driving factors behind certain metrics. Additionally it contained metrics for metric-based monitoring of the systems rather than the projects as well. Even though we and the metric expert team at GDIS found the GDIS Metric Model to be a valuable resource it was fairly unknown by the project managers. This was mostly due to access problems and access inflexibility to the metric specifications. Additionally, GDIS did not provide a template for the project specific tailoring of metrics or reports. Hence, the (project specific) tailoring of the metrics was not documented and unknown to the metric expert team.
- GDIS already implemented a CMMI level 3 compatible measurement and analysis process that was based on the ISO 15939 measurement process [11]. This process is routed in a standard PDCA cycle and contains evaluation of the metrics and the metric process as well.

The details above are summarized in the following central findings:

1. A lot of heterogeneous data sources need to be integrated to provide a sound basis to compute metrics.
2. Data source integration works but could be better.
3. There are no M² dashboards for project management (PM M² dashboard for short).
4. The documentation of the metrics is good, but hard to read, hard to understand and hard to access.
5. The metric process is well defined and suitable.

B. 2011 - Project Management M² Dashboard

In 2011 we started to address the (from our point of view) important points 3 and 4 of the list of findings. The following sections briefly describe our approaches and ideas.

1) Development of a template for PM M² Dashboard

Together with experts from Kugler Maag CIE we started to develop a PM M² dashboard early in 2011. We strongly believe and our experience shows that it is important to align the PM M² dashboard with the information needs of the metric customers. Hence, we started the development with gathering information needs from the project managers. The gathering was realized by several interviews with different project managers. We then consolidated the requirements to an initial set of information needs which should be answered by the PM M² dashboard.

We then started with the incremental development of the PM M² dashboard template. In every increment we included one new monitor as a paper prototype. These paper prototypes are evaluated in a workshop with the group as well as with project managers.

The basis for the development of the PM M² dashboard template was a template from Kugler Maag CIE developed in Excel. This was then tailored to fit the prototypes developed in the stage before. The PM M² dashboard itself, however, was still implemented as an Excel spreadsheet. The benefit of this approach was that the project managers could easily tailor this spreadsheet to fit their individual needs. A problem on the other hand was the huge development effort to integrate the

[1] http://www-03.ibm.com/software/products/us/en/reqpro/

[2] http://www-03.ibm.com/software/products/us/en/clearquest/

[3] http://www.ibm.com/developerworks/rational/products/rsa/

[4] http://www.tricentis.com/en/solutions

Software Engineering Conference (APSEC), 2013 20th Asia-Pacific , vol.2, no., pp.73,78, 2-5 Dec. 2013
doi: 10.1109/APSEC.2013.115

different data sources in the spreadsheet. Later on we also experienced severe maintenance problems with the complex VBA scripts which realized the data aggregation and data integration mechanisms.

The project managers who participated in the development of the PM M² dashboard template were then picked to evaluate it out in the field. This pilot phase was important to tweak the details of the PM M² dashboard implementation and to test the data aggregation and data integration mechanisms. It was also important to investigate the benefits of the dashboard for the project managers in the field before mandatorily requiring every project manager to use the PM M² dashboard for their project monitoring and reporting.

Together with the PM M² dashboard template we developed a metric specification sheet describing the metrics and monitors in the PM M² dashboard. These sheets were also used to document project specific tailoring of the PM M² dashboard. Because some tailoring is inevitably (like reporting, timing intervals, data sources, and so on) most of the projects kept individual specification sheets for their PM M² dashboard. These specifications where stored in Excel spreadsheets, one sheet per metric specification. The specification attributes were aligned with CMMI level 3 requirements for metrics and analysis.

2) Design and Construction of metric documentation tools

We believe that understandable and accessible documentation which is oriented at the information needs of the metric customers is another important aspect to the success of the PM M² dashboard. An important aspect in our scenario was the replacement of the existing metric documentation - the GDIS Metric Model. This documentation already contained important information about the metrics as well as visualization prototypes. It also mentioned important questions and aspects which could be answered by using a specific monitor. The new documentation should retain this information yet overcome the weaknesses of a large inflexible Word document.

During the design of the documentation we realized that we need to address two different stakeholders. The metric customer who seeks information about specific metrics and the metric expert who needs to specify the metrics.

We decided to address the metric customer with a special HTML-based metric documentation that allows easy navigation between the different aspects of the metrics and monitors. Additional pages which contain a catalogue of categorized information needs help the metric customers to find specific monitors and metrics based on their needs. We also added interpretation aids to the monitor and metric documentation which help metric customers to interpret monitors that they only need to analyze occasionally.

The metric experts required a tool for easy documentation of the metrics and monitors. Due to the specific documentation requirements, we built a specialized tool based on a metric and monitor meta-model. The model-driven approach resulted in fast prototypes for the solution. It also allowed us to develop the tool and meta-model incrementally and iteratively. The HTML-based metric documentation was generated using model-to-text transformations.

C. 2012 - SE-Quality Indicators and Maintenance of PM M²-Dashboard

In 2012 again two main activities occupied the metric expert team at GDIS. The first was the development of a set of metric-based monitors (so called Software Engineering Quality Indicators) to monitor the overall quality of the software development at GDIS from a management perspective. The second was the continuous maintenance and extension of the PM M² dashboard.

1) SE Quality Indicators

As the SE quality indicator were developed internally at GDIS, we just want to present the outcome of the development process: the SE-quality PowerPoint slides as well as the data integration and measurement mechanisms. The goal of the slides as already mentioned is to provide management with an aggregated compact and condensed set of metrics to monitor the overall performance of the software engineering part of the company. The base data for these quality indicators is taken mostly from the project data sheet as well as other sources that are sometimes also used in the PM M² dashboard. The SE quality indicators are calculated in a dedicated business information tool (IBM InfoSphere DataStage[5]), contrasting the data integration mechanism in the PM M² dashboard, which is based on Excel spreadsheets. However, yet again Excel is used to visualize the monitors for the quality indicators. These visualizations are then embedded in a PowerPoint slideshow and handed over to the managers.

2) Maintenance of the PM M² dashboard

In 2012 the PM M² dashboard became a mandatory tool for the project managers. Most of the metrics were part of the quality control cycle of the project and needed to be reported to the steering committee of the project for example. Due to this we were faced with maintenance problems because of the complex data aggregation and integration mechanisms under the hood of the spreadsheet for the PM M² dashboard. We also constructed additional monitors to fulfill new information needs from the project (and test) managers. These monitors are mostly based on test specification metrics but we also added dedicated monitors for newly updated risk and open point lists.

These new monitors got then included in the metric documentation as well. In sync with this change we also altered the specification spreadsheets for project specific tailoring of the metrics in the dashboard. The new sheets now directly point to the metric documentation and only contain the tailoring but not the complete specification. This also required hard versioning of the metric documentation because the tailoring sheets needed to refer to a specific version rather than the newest version of the documentation.

[5] http://www-03.ibm.com/software/products/us/en/ibminfodata/

doi: 10.1109/APSEC.2013.115

D. 2013 – Transition

The year of 2013 brought a lot of organizational changes to the company. Alongside with these changes new tools were introduces over the last 2 years. These changes need to be reflected in the PM M² dashboard of course.

To adequately address these changes we started to gather changed requirements for a renewed version of the PM M² dashboard. We gathered the requirements by performing several interviews with different project managers. The consolidated new requirements then showed a lot of new information needs and also showed that many of the information needs that were addressed with the old template for the PM M² dashboard were not so important anymore. This inevitably led to a complete rework of the template. This also showed that it is very important to constantly (re) check if a PM M² dashboard template is addressing the *right* questions. We only started this over two years after the initial requirements gathering which proved to be too much time between these checks.

Due to the heavy maintenance and usability problems with the old Excel-based template of the PM M² dashboard we started to rethink the measurement and integration infrastructure for the PM M² dashboards at GDIS. We are currently implementing a web-based service oriented federalist measurement infrastructure to overcome these problems [12]. We plan to start using the infrastructure in first pilots until the end of the year. We decided not to use a BI-based system (like the SE quality indicators) because we anticipate heavy continuous changes to the infrastructure, the metrics, the monitors and the tools that need to be adapted. These changes are hard to address in a centralistic solution that requires a consolidated data model. Our federalist infrastructure that is built using loosely coupled measurement and calculation services is a lot more flexible in these scenarios.

IV. Lessons Learned

This section concludes the lessons learned from the history of the metrics program at GDIS. We list the different lessons learned one after another and provide a general conclusion at the end of the section.

A. Companywide complex metrics programs require a solid measurement infrastructure

The history above shows various maintenance problems resulting from a measurement infrastructure that was developed over time. These problems include:

- The Excel-based integration leads to stiff adaption of new data sources.
- Debugging and development of the multiple Excel sheets is hard and sometimes impossible.
- There is o real time data available in the PM M² Dashboards due to complex interaction between the different sheets and tools as well as required timings for the source data.

As we already mentioned above the different stakeholders have different requirements to the measurement infrastructure. The following subsections investigate these requirements in more detail.

B. The M² dashboard design is important for a long time success of the measurement iniative

Unfortunately, the design of the actual M² dashboards is often not addressed with the importance that it needs. Most of the time the M² dashboard design follows the templates provided in Excel or PowerPoint. However, together with the wide pallet of possibilities provided by these tools this often leads to M² dashboards that are full of eye candy (different bright colors, stacked diagrams, multiple font styles, 3D effects, etc.). Even though these dashboards may look impressive at first these effect dramatically reduce the everyday usability of the M² dashboard because an observer is distracted by it and hence it is hard to focus on the important aspects that need attention. Stephan Few provides a rich set of design guidelines for effective dashboards and the visualization of numbers [13], [14].

C. The needs of different (metric) stakeholders need to be addressed (differently)

A metrics program that influences the whole company needs to address the needs from different stakeholders for the metrics program. From our experience we can extract (at least) four different and distinct roles:

- Measurement Customer
- Metric Expert
- Metric Developer
- Metric Infrastructure Operator

Each of these stakeholders provides a unique and specific set of requirements regarding the architecture and the provided functionality of the underlying measurement infrastructure.

1) Measurement Customer

A project manager is a typical example of a measurement customer. The PM is interested in the actual status of the project and does not care (and should not!) about the (technical) way the data is collected or the technical details of the metric calculation. Measurement customers have a brought variety of information needs. Unfortunately, the answers to the different information needs are stored in various repositories. Typical scenarios include different systems for budget, scheduling, tasks and risk information. The resulting central requirement of the measurement customer for a PM M² Dashboard and its underlying measurement infrastructure is the integration of these systems in a way that a comprehensive calculation of metrics is possible.

Additionally, measurement customers demand correct and up-to-date data because old or incorrect data may lead to wrong conclusions and wrong decisions. Hence, the measurement infrastructure should provide mechanisms that guarantee a fast recognition and processing of relevant events

Software Engineering Conference (APSEC), 2013 20th Asia-Pacific , vol.2, no., pp.73,78, 2-5 Dec. 2013
doi: 10.1109/APSEC.2013.115

inside the system landscape. Additionally, this requires a robust and highly availability measurement infrastructure.

The GDIS metrics initiative shows that the information needs of measurement customers often change over time. For example development tools and systems are replaced by other tools or systems (tool evolution). Of course, the new tools and systems need to be integrated in the infrastructure. Additionally, processes and organization schemas of enterprises often evolve as well. Especially reorganizations lead to new and changed responsibilities of individual measurement customers and roles which inevitably lead to changes in information needs. Concluding from this, an important requirement for a measurement infrastructure is to support the evolution of metrics, integrated systems, and visualizations.

2) Metric Expert

The metric experts are supervisors of the metric processes and provide metric solutions for the measurement customers. They design the measurement infrastructure and provide M² dashboard templates as well as templates for the monitors. They specify the metrics and monitors to provide a solid base for the development of metric services, dashboards and infrastructure components by the metric developers.

3) Metric Developer

The developer needs to implement metrics, visualizations and tools to gather data. The measurement infrastructure needs to support the developer with a clear structure and concepts for all specific tasks. The task to integrate a new system into the infrastructure to gather its data is completely different from the implementation of a new metric calculation algorithm or the implementation of a new visualization. Hence, a requirement for the infrastructure is clear separation of system integration, calculation, and visualization.

4) Metric Infrastructure Operator

This role is often ignored while designing and building a measurement system or measurement infrastructures. We ignored this role in the past as well. Unfortunately this leads to a situation where everybody is part of the operations team to a certain degree.

Typically the operations department has two main responsibilities. First, it has to guarantee that all systems are working inside their operational parameters. This requires a dedicated set of operation tools as part of the infrastructure. The measurement infrastructure should at least provide or support a monitoring tool which allows analyzing the amount of data that is transported and stored in the components of the measurement infrastructure. Second, the operations department has to solve upcoming problems in the infrastructure without disturbing the integrated systems as these systems are often of crucial importance for the company. The operation department is also responsible for the alignment of the system landscape of the company. Hence, the infrastructure should be compatible with service oriented architectures found in modern companies.

D. Starting and maintaining a companywide metrics program is hard and time consuming

The metrics program at GDIS is still in development even though we are now thoroughly working on the topic for over 4 years. This shows a typical time span of such programs in large companies. Another fact is that so far we only developed M² Dashboards for the project managers and test managers. The general management is also addressed by the means of the SE-Quality Indicators. However, there are many more roles that require metric-based monitoring. For example we could also address: software-architects, integration-managers, configuration-managers or developers. This is still a lot of work ahead of us.

E. Continuous evaluation of the M² Dashboard and their templates is important

We explained the initial development of the PM M² dashboard in the history above. The resulting PM M² dashboard suited the needs of the project managers nicely due to their participation during the development. When we evaluated the PM M² dashboard requirements in 2013, however, we noticed that most of the project managers required new or different monitors in their PM M² dashboard. This was due to the long time span (over two years) between the initial requirements gathering and our requirement evaluation in 2013.

V. Best Practices for a Companywide CMMI Level 3 compatible Metrics Program

This section contains the consolidated best practices from the lessons learned discussed above. We organized the best practices in two categories *organization and M² dashboard development* and *measurement infrastructure* because we believe these provide the most benefit for all stakeholders. Hence, they form the backbone of a successful metrics program.

A. Organization and M² dashboard development process

This process component is often ignored. Even though metric based monitoring initiatives are launched in companies that apply good development processes and best practices for the development of high quality and successful software they seem to ignore all of this when it comes to the development of the metrics and measurement systems. Hence, the following list provides a condensed set of organizational best practices that we found to be the most important ones:

BP-O1. Clearly define the roles and responsibilities inside the metrics expert team.

BP-O2. Include metric developers and operation personnel in the development as well.

BP-O3. Continuously evaluate the information needs of the metric costumers.

BP-O4. Develop different M² dashboards to fit the different roles that have different information needs.

Software Engineering Conference (APSEC), 2013 20th Asia-Pacific , vol.2, no., pp.73,78, 2-5 Dec. 2013
doi: 10.1109/APSEC.2013.115

BP-O5. Use software engineering best practices (like prototyping and iterative and incremental development) for the development of metric and measurement systems.

These best practices are best addressed by an iterative and incremental M² dashboard development process. We plan to publish a detailed paper on the concrete development process that we designed and used in the near future.

B. Measurement Infrastructure

Regarding the measurement infrastructure we identified the following best practices:

BP-I1. Support the easy integration of heterogeneous systems to provide the basis for different metrics and visualizations.

BP-I2. Enable the fast and up-to-date recognition and update of the metrics on a change in an integrated system.

BP-I3. Clearly separate system integration, metric calculation and visualization.

BP-I4. Be robust to avoid a complete system failure if a small part of the system fails. Additionally, the failure of the infrastructure should not result in a failure of the integrated systems.

BP-I5. Do not use one central database to store the measurement values.

BP-I6. Do not develop one central data schema to avoid schema-mapping problems.

BP-I7. Support the evolution of metrics, integrated systems, and visualizations.

BP-I8. Offer dedicated operation tools.

BP-I9. Be compatible to Service-Oriented-Architectures.

This set of measurement infrastructure best practices will inevitably lead to a loosely coupled federalist infrastructure like we proposed in [12].

VI. Conclusion

In this paper we described the history of the metrics and measurement initiative at Generali Deutschland Informatik Services over the last 4 years. We extracted lessons learned from our experience and provided a consolidated list of best practices for measurement and metric initiatives. Currently, we are developing new templates, monitors and metrics. Furthermore we are trying to move from the current Excel-based metric-based measurement dashboards to a web-based service oriented solution to overcome maintenance and usability problems with the existing solution.

References

[1] C. P. Team, “CMMI® for Development, Version 1.3 CMMI-DEV, V1.3,” 2010.

[2] V. R. Basili, G. Caldiera, and D. H. Rombach, “Goal Question Metric Paradigm,” in in *Encyclopedia of Software Engineering*, vol. 2, no. 1, J. J. Marciniak, Ed. Wiley-Interscience New York, NY, USA, 1994, pp. 528–532.

[3] L. Cyra and J. Górski, “Extending GQM by Arguement Structures,” in *CEE-SET 2007*, 2008, vol. 44, no. 5, pp. 26–39.

[4] J. Münch and J. Heidrich, “Software project control centers: concepts and approaches,” *Journal of Systems and Software*, vol. 70, no. 1–2, pp. 3–19, 2004.

[5] J. Heidrich and J. Münch, “Goal-oriented setup and usage of custom-tailored software cockpits,” in *PROFES ’08: Proceedings of the 9th international conference on Product-Focused Software Process Improvement*, 2008, pp. 4–18.

[6] E. Palza, A. Abran, C. Fuhrman, and E. Miranda, “V & V Measurement Management Tool for Safety-Critical Software,” in *Proceedings of IWSM/MetriKon 2004*, 2004.

[7] M. de Los Angeles Martin and L. Olsina, “Towards an Ontology for Software Metrics and Indicators as the Foundation for a Cataloging Web System,” in *Web Congress*, 2003, pp. 103–113.

[8] L. Westfall, “12 Steps to Useful Software Metrics.” The Westfall Team, 2005.

[9] P. Berander and P. Jönsson, “A goal question metric based approach for efficient measurement framework definition,” in Proceedings of the 2006 ACM/IEEE international symposium on International symposium on empirical software engineering - ISESE ’06, 2006, p. 316.

[10] F. T. Anbari, “Earned value project management method and extensions,” *IEEE Engineering Management Review*, vol. 32, no. 3, pp. 97–97, 2004.

[11] ISO/IEC 15939: 2002 “Software Engineering - Software Measurement Process.”2002.

[12] M. Vianden, H. Lichter, and A. Steffens, “Towards a Maintainable Federalist Enterprise Measurement Infrastructure,” in Joint Conference of the 23nd International Workshop on Software Measurement (IWSM) and the 8th International Conference on Software Process and Product Measurement (Mensura), 2013.

[13] S. Few, *Show Me the Numbers*. 2012, p. 343.

[14] S. Few, Information Dashboard Design: The Effective Visual Communication of Data. O’Reilly Media, Inc., 2006.

Software Engineering Conference (APSEC), 2013 20th Asia-Pacific , vol.2, no., pp.73,78, 2-5 Dec. 2013
doi: 10.1109/APSEC.2013.115