Implementing a Software Metrics Program at Nokia

Tapani Kilpi, *Nokia*

Metrics play an important role in many software organizations' continuous process improvement activities. In an organization with mature software processes such as Nokia, carefully adjusting the measurement program to the environment-specific needs can save considerable effort.

All software organizations experience chaos. In theory, the solution is easy: define and implement all key processes clearly enough, and the chaos is gone. Software process improvement is an approach for defining, organizing, and implementing software processes. Well-known SPI methodologies include ISO/IEC 9000 standards,[1] the Capability Maturity Model,[2] Bootstrap,[3] and ISO/IEC 15504.[4] Measuring plays a large role in these methodologies; to address this, an organization will normally implement measurement programs,[5] such as the Goal-Question-Metric method.[6] However, organizations at varying maturity levels do not always need to follow the same steps. Focusing on effort usage—the working hours spent on implementation—this article will introduce and outline the Nokiaway metrics program, show how it differs from GQM, and explain the differences.

Nokiaway

Nokiaway is a real-world method used in Nokia's Fixed Switching Research & Design Department. FSG R&D, named hereafter FSG, produces software for the DX200 product line of telephone switches. The DX200 program library has over 15 million source lines of code, and Nokia has committed more than 2,000 engineers to carry out DX200 development work. DX200 is possibly Finland's largest software project.

The Nokiaway metrics program comprises these continuous metrics activities: metrics definition, data collection, metrics analysis, and metrics reporting.

Metrics definition

In FSG, metrics definition is a part of continuous process improvement—that is, defining the metrics follows the same organizational structure as process improvement does. At the highest organizational level, the FSG management team steers FSG's process improvement (see Figure 1). FSG's processes comprise five separate process areas: release, design, testing, maintenance, and quality assurance. The process teams, each of which is responsible for a specific process area, manage these areas. A process development manager—a full-time process developer—heads each process team. Other team members are

0740-7459/01/$10.00 © 2001 IEEE

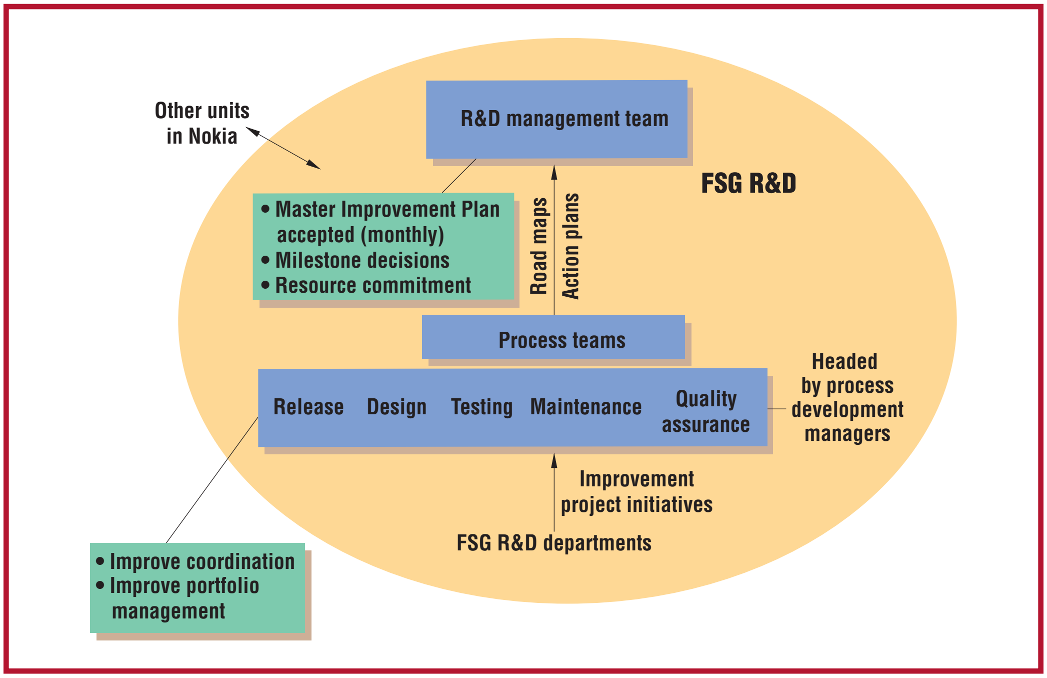

Figure 1. The FSG process improvement steering structure.

part-time process developers whose primary responsibilities include software development, testing, and so on.

The FSG software quality assurance (SQA)[7] process team has the overall responsibility of coordinating and running the measurement program. Other process teams are responsible for planning detailed improvement projects and defining the metrics belonging to their process area. The management team defines the long-term process improvement road maps and action plans and accepts and gives resource commitments to process improvement projects that the process teams propose. The FSG R&D departments implement the improvement projects.

Typically, a process team defines and continuously updates a set of 10 to 20 metrics. Each metric addresses one of FSG's approximately 20 metric profiles (including Release Program, Sw Project, Testing Project, Department, and Section). For instance, each process team has added one to three metrics to the Sw Project metrics profile. Many metrics are common to several different metrics profiles, but no profile resembles another. Typically, a metrics profile contains approximately ten metrics that most of the process teams (not necessarily all of them in all cases) define. The different process teams collectively define the sets of metrics for the FSG metrics library.

Data collection

Ideally, data collection should be set so that the metrics save (that is, collect) data at its inception. This requirement demands a relatively high level of process maturity. The Nokiaway metrics program uses four different management tools and their data storage for collecting and creating metrics: Resource and Project, Fault, Test Case, and Inspection and Review. FSG project managers, software designers, test engineers, fault assistants, and people in other corresponding roles routinely use these tools. From the metrics point of view, this arrangement provides excellent circumstances for data collection. No one must collect the data after inception, because it already exists in data storage for each of the four tools.

For example, project managers use the Resource and Project tool to control projects, so they are eager to feed in the required data. Other people in the project must also feed in their data, but it's such a simple task that no one really complains. We similarly apply the other tools: they are a part of daily processes, and we feed data to them. Quality engineers then have the relatively easy task of producing the metrics values by using the management tools. This minimizes the traditional manual hard work, and we can direct attention to analyzing the metrics values.

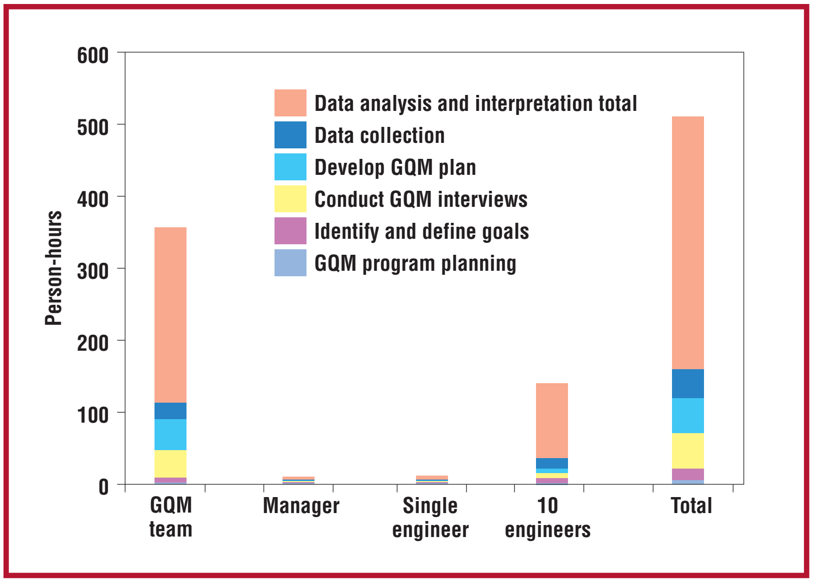

Figure 2. An example effort model for a routine application of GQM.[8]

Metrics analysis

In many real-life metrics programs, metrics analysis gets little attention while other metrics activities are handled reasonably well. However, people in such organizations probably seldom ask themselves, "Who should react to the metrics values, and who should ensure that the necessary corrections are carried out?" If no one uses a metric, there's no point in producing it.

Defining analysis guidelines and analyzing responsibilities for metrics play a vital role in FSG metrics definition. This ensures that someone, with some instructions, analyzes a metric's results. In FSG, SQA people and the quality engineers generate the metrics reports. They have been trained to analyze the metrics values based on each metric's analysis guidelines. Typically, a quality engineer produces a set of approximately 10 metrics and analyzes them within the project's scope. This quality engineer also has the expertise to carry out the local analysis of a metric's value. In FSG, we could say that a quality engineer's metrics duties focus primarily on analysis instead of data collection. This is not the traditional way, is it?

Metrics reporting

The metrics and analysis results are often all that most people in an organization see from a metrics program. These reports are typically saved to a database or published on the Web. From both of these places, everyone in the organization can access the reports—in theory. What's missing is the periodical review carried out by predefined people. Also missing are the decisions these people make on the basis of the metrics results. In a situation such as this, there is no hope that changes are planned and carried out on the basis of the metrics.

The FSG metrics profiles are defined to measure different types of targets. If the measured target is, for example, a project, the project's quality engineer produces the Project profile metrics. Then, the quality engineer presents the metrics report at a project meeting. This works well when the quality engineer knows the project and people involved. This makes it easier to translate the metrics values to an understandable language and communicate the results to those who should be most concerned with the metrics values: the people in the project.

A comparison of GQM and Nokiaway

GQM is a well-known and widespread method for planning and implementing a metrics program.[7] It represents a systematic approach for tailoring and integrating goals to the software process models, software products, and quality perspectives of interest—on the basis of the project's and organization's specific needs. (FSG defines the corresponding goals as part of the Master Improvement Plan, a yearly process improvement plan.) I compare Nokiaway with the GQM as exemplified by Rini van Solingen and Egon Berghout's practical interpretation and example effort calculations of implementing a GQM metrics program.[8]

General characteristics

Nokiaway is a free interpretation of GQM. With respect to a standard GQM approach, it has these essential enhancements:

- It uses a quality metrics library instead of defining a new set of metrics for each project.
- It automatically, instead of manually, collects raw data for the metrics.
- It semiautomatically (instead of manually) produces the reports' metrics charts.
- Most people running operative tasks in the measurement program are part-time instead of full-time employees.
- It uses the Quality Plan and FSG Met-

rics Guidelines instead of a GQM plan for defining a measurement program's exact background and scope.

A fundamental reason for these extensions is that FSG is a multiproject environment; that is, several projects of the same type run in FSG instead of just one project. It is important that the experiences of all FSG's measurement programs are collected and analyzed to further improve the metrics. FSG saves this data in the FSG Metrics Guidelines; that is, the document's contents form a metrics library.

Some GQM purists might criticize Nokiaway for depending too much on predefined metrics and losing sight of the GQM idea of defining goal-driven metrics separately for each specific case. I have this response:

- Is there really no point in defining common metrics that can be applied across several similar projects in an organization?
- Do independently defined metrics for different but similar projects really vary that much from each other?
- Doesn't management, after all, have the main responsibility in deciding an organization's process improvement strategy—including goals?
- Could an organization provide individuality in different measurement programs by selecting a subset of metrics from a predefined, actively updated metrics library?
- Aren't collecting raw metrics data and producing the metrics reports laborious tasks, which require exhaustive resources when performed manually?

While posing these questions with certain answers in mind, we must remember that FSG is a large organization with specific characteristics. Environment and circumstances greatly influence planning an organization's measurement program. My hope is that FSG's experiences can offer other large organizations something to think about when applying GQM or another measurement program method.

Effort models

Solingen and Berghout presented an example effort model (see Figure 2) for the routine application of a typical GQM measurement program with these characteristics:[8]

- Measurement program target: one software project.
- Project team size: 11 persons (one project manager and 10 software engineers).
- Project duration: approximately one year.
- One major goal.
- An existing tool infrastructure.
- No need for special training; the participants are familiar with the program.
- Feedback sessions during the project: five.

Figure 2 also shows the following cost structure, which is typical for a routine GQM application:

- Roughly 30 percent of the effort goes toward defining the measurement program, and 70 percent goes toward continuing the measurement program, which is almost completely spent on feedback sessions.
- On the measurement program, the GQM team spends 70 percent of the effort; the project team spends only 30 percent of the total effort.
- The project team's effort is less than 1 percent of their total working time.
- A typical GQM measurement program over one calendar year in a project of 11 persons requires three person-months of effort.[8]

Table 1 compares GQM and Nokiaway activities, with Solingen and Berghout's description of their effort model in the left column.[8] In GQM, the GQM plan contains the detailed description of the measurement program's activities, their schedule, and their associated responsibilities. In FSG, the quality plan contains the corresponding information for implementing a measurement program.

Figure 3 presents the total resource use in routine applications of standard GQM and Nokiaway. (Analyzing the realized hours of 10 FSG projects has produced the estimations for the realized working hours spent on running a measurement program in FSG.) The total effort in Nokiaway appears to be less than that of the GQM measurement program's. This is because

In many real-life metrics programs, metrics analysis gets little attention while other metrics activities are handled reasonably well.

Table 1

Correspondences between the Goal-Question-Metric and Nokiaway activities

Goal-Question-Metric[8]	Nokiaway
■ *Plan GQM program.* This includes identifying available input, preconditions, and constraints; setting up an infrastructure; selecting an improvement area and a project; initially planning the measurement program; and preparing and training the project team.	■ *Define FSG metrics guidelines.* This document defines all quality metrics that FSG uses, their intended use, directions and criteria for analyzing them, and so on. ■ *Prepare a quality plan.* This includes definitions for quality organization and responsibilities, a schedule for quality feedback sessions, training activities, most common tools and instructions, internal and external verification activities, quality criteria and metrics (a subset selected from the FSG metrics guidelines), a schedule and target group for quality reports, and a status control table for the quality plan follow-up.
■ *Identify and define GQM goals.* This includes characterizing projects and organizations, identifying and selecting improvement goals, defining measurement and GQM goals, modeling relevant software processes, and identifying artifacts for reuse.	■ *Define measurement key areas.* FSG process teams define the process areas' process improvement goals on the basis of their observations and on the quality metrics used to evaluate these areas.
■ *Conduct GQM interviews.* This includes studying documentation; identifying, scheduling, and inviting interviewees; briefing project teams; and conducting and reporting GQM interviews.	■ *Apply metrics feedback.* All quality metrics that FSG uses have been defined in close cooperation with FSG practitioners. The process teams and the SQA people collect the feedback between projects to improve the metrics for the next use.
■ *Develop GQM deliverables.* This includes defining, reviewing, and refining a GQM plan; defining a measurement plan; identifying and defining data collection procedures; reviewing and refining a measurement plan; and developing an analysis plan.	■ *Update the FSG metrics guidelines.* ■ *Prepare the quality plan's measurement parts.* This part of the quality plan defines a subset of the metrics included in the FSG metrics guidelines according to the project-specific process improvement goals and possibly defines project-specific directions for applying the metrics.
■ *Collect data.* This includes a trial to test data collection procedures and forms; briefing the project team; launching the measurement program; and collecting, validating, coding, and storing measurement data. ■ *Analyze and interpret data.* This includes analyzing the measurement data; preparing the presentation material; and planning, conducting, and reporting the feedback sessions.	■ *Collect data.* Process-oriented tools automatically collect all data needed for producing the FSG quality metrics. ■ *Analyze and interpret data.* The FSG SQA personnel carry this out; most of them are software or test engineers working in SQA part-time.

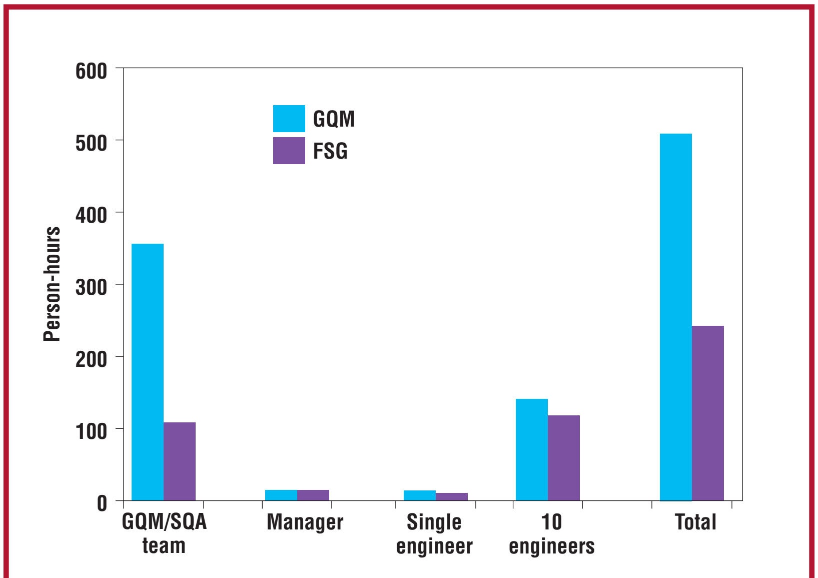

Figure 3. Effort usage comparison between GQM and Nokiaway.

Nokiaway uses a high degree of automation in data collection while GQM does not. However, because we can apply automated data collection to GQM, the overall Nokiaway effort profile differs insignificantly from an average GQM effort profile.

A well-planned and carefully focused measurement program is a useful aid for an organization that highly values continuous process improvement. Some capable methods, such as GQM, for planning and implementing a measurement program tend to offer an independent and complete set of steps and means for process improvement. There is nothing wrong in this, but many organizations already have a successful history in applying different approaches, such as ISO/IEC, CMM, and Bootstrap. As a result of this evolution, these organizations have already developed their processes to a relatively high maturity level and do not need to start over.

The enhancements to GQM I've described can save approximately 50 percent of the cost of implementing and running a measurement program that relies on manual work. These improvements also help minimize bureaucracy and prevent people from losing their motivation for measuring.

References

1. *ISO 90001*, *Quality Systems—Model for Quality Assurance in Design/Development, Production, Installation and Servicing*, Int'l Organization for Standardization, Geneva, 1994.
2. B. Curtis, W.E. Hefley, and S. Miller, *People Capability Maturity Model (P-CMM)*, tech. report CMU/SEI-95-MM-02, Software Eng. Inst., Carnegie Mellon Univ., Pittsburgh, 1995.
3. P. Kuvaja et al., *Software Process Assessment and Improvement: The Bootstrap Approach*, Blackwell, Oxford, UK, 1994.
4. *Information Technology: Software Process Assessment*, ISO/IEC tech. report 15504 Type 2, Int'l Organization for Standardization and Int'l Electrotechnical Commission, Geneva, 1998.
5. N.E. Fenton and S.L. Pfleeger, *Software Metrics: A Rigorous and Practical Approach*, Int'l Thomson Computer Press, London, 1996.
6. V.R. Basili, C. Caldiera, and H.D. Rombach, "Goal Question Metric Paradigm," *Encyclopedia of Software Engineering*, vol. 1, J.J. Marciniak, ed., John Wiley & Sons, New York, 1994.
7. *ISO 9000-3*, *Guidelines for the Application of ISO 9001 to the Development, Supply and Maintenance of Software: Quality Management and Quality Assurance Standards—Part 3*, Int'l Organization for Standardization, Geneva, 1991.
8. R.van Solingen and E. Berghout, *The Goal/Question/Metric Method: A Practical Guide for Quality Improvement of Software Improvement*, McGraw-Hill, Cambridge, UK, 1999.

About the Author

Tapani Kilpi is the chief researcher at the Solid Applied Research Center, Oulu, Finland. His current responsibilities focus on network management and 3G–4G applications. He received his MS and PhD in information processing science at the University of Oulu. Contact him at Solid, Elektroniikkatie 6, 90570 Oulu, Finland; tapani.kilpi@solidtech.com.

For more information on this or any other computing topic, please visit our Digital Library at http://computer.org/publications/dlib.