# GQM based Inspection

Erik van Veenendaal / Mark van der Zwan

## Abstract

*Inspections are generally accepted as a means to improve the quality of software products in an effective and efficient way. However, inspections are not a standard practice in a great number of software projects and software organisations. Introducing and implementing inspections is often a tedious and difficult task, because software engineers must be personally convinced of the effectiveness of new methods before they will consistently use them.*

*Collecting relevant data during inspections is a way to overcome these difficulties. Such data collection for software inspections is termed measurement. Measurement is a powerful aid to implement and improve the inspection process. Showing real-life data is often convincing for both the software engineers and their managers. A well-known and popular software measurement approach is the Goal/Question/Metric method (GQM). Applying GQM to the inspection process helps to focus the data gathering process, and support the interpretation process. An important part of the measurement programme and thus inspection implementation and improvement process are the so-called feedback sessions. Feedback sessions are meetings involving members of the project team and the measurement team. It is an essential tool for analysis and interpretation of the measurement results.*

*The background to this paper is the implementation of inspections in a number of Dutch organizations using the GQM approach as a main vehicle. Practical examples are provided of the measurement goals, metrics and feedback sessions.*

## 1. Software Inspections

In every software development phase defects are introduced, found and rework is being carried out. However, often most defects are only found when the software product is almost finished, e.g. during the system and acceptance testing phase, or even during operation. Defects found during the testing phase have the disadvantage that their rework on the almost finished software product is very time consuming. It would have saved the development organisation a lot of time if these defects where found during an earlier development phase.

Inspections are an effective and efficient measure that can be introduced to improve the quality of the products at an early stage [2]. Besides finding a defect at the earliest possible moment, the prevention of defects is the important issue. Inspections can also be used as a means for defect prevention. Based on an analysis of the defects that were found, the software development processes can be adapted and optimised to prevent these defects from occurring in the future (as far as possible). Engineers that are involved in the inspection process can learn from their defects or the defects that were made by someone else. Inspections can be defined as a structured review of an engineers' software work product carried out by his colleagues to find defects and to enable the engineer to improve the quality of the product [1].

While the importance and benefits of inspections for software projects is well understood within the software industry, only few engineers apply the inspection technique to their personal work. Even when statistic evidence from other organizations and projects exists [4] [7], the introduction of improved software methods, e.g. inspections, is often slow because software engineers must be personally convinced of the effectiveness of new methods before they will consistently use them. In software this is particularly true because [3]:

- software engineers' methods are largely private and not obvious from the products they will produce. Thus, if they do not use proper methods, it is unlikely that anyone else will know;
- software engineers are generally not trained to follow the planning and measurement disciplines needed to rigorously evaluate the methods they use;
- even when software groups have a common set of defined practices, these practices are not consistently followed;
- the current industrial environments do not as a prerequisite require the use of the best-known software engineering methods.

A principal issue, therefor, is how to motivate and implement inspections within a software organisation. The authors argue that should metrics play a major role in convincing both the software engineers and their management and tuning the inspection process. In fact metrics are a critical success factor to successful inspection implementation. Metrics are not optional, they are a requirement.

## 2. Goal/Question/Metrics approach

A well-known and popular software measurement approach is the Goal/Question/Metric approach [5] [6]. GQM represents a systematic approach to tailor and integrate goals with software process and products models. It is based on the specific needs of a project and organisation. Within GQM measurement goals are derived from high-level corporate goals, and further refined into measurable values (metrics). GQM defines a certain goal, refines this goal into questions, and defines metrics that must provide the information to answer these questions. The GQM paradigm provides a method for top-down metric definition and bottom-up data interpretation (figure 1).

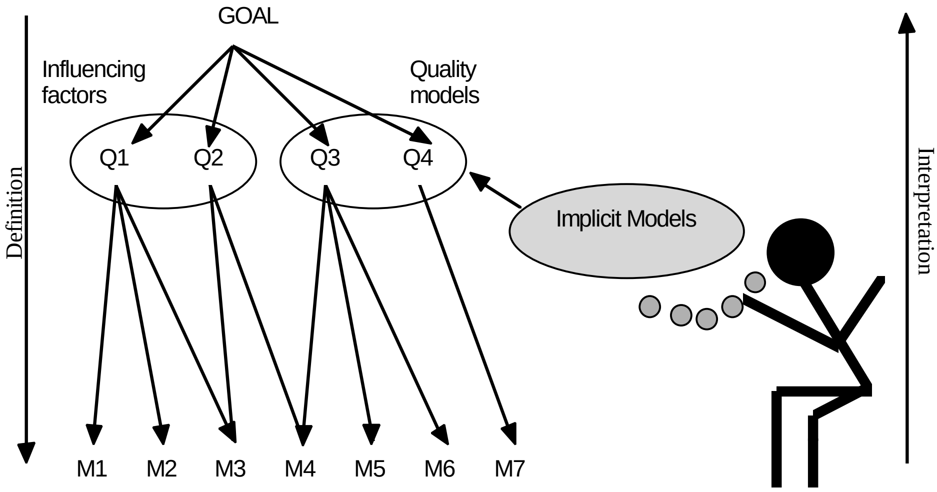


*Figure 1 : The Goal Question Metric paradigm*

A number of steps can be distinguished with the GQM process. The steps to take are applicable when introducing (inspection) measurement in practice. Figure 2 shows these steps.

*Step 1*: Organisation and project characterisation. Defining measurement programmes starts with a characterisation of the organisation and the project. The results of the characterisation are used in the definition, ranking and selection of the goals and also in establishing the GQM-plan.

*Step 2*: Goal definition. The second step in the GQM process is defining measurement goals. Goals can directly reflect business goals, but also specific project goals or personal goals. Measurement goals must be carefully selected, based on selection criteria such as: priority to the project or organisation, risk, time in which a goal can be reached.

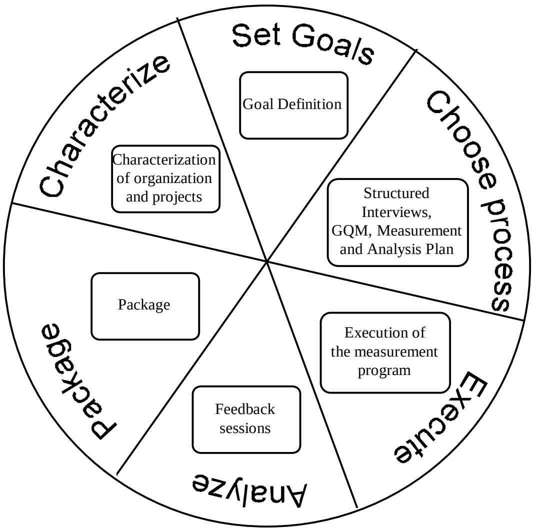

Set Goals
Goal Definition
Choose process
Structured Interviews, GQM, Measurement and Analysis Plan
Execute
Execution of the measurement program
Analyze
Feedback sessions
Package
Package
Characterize
Characterization of organization and projects

*Figure 2 : The steps for goal-oriented measurement*

*Step 3*: Developing the measurement programme. The major activity when developing a measurement programme is refining the selected goals into questions and metrics. It is important to check whether metrics answer questions, and that answers to such questions provide a contribution towards reaching the defined goals.

*Step 4*: Execution of the measurement programme. In the execution step of the measurement programme, the measurement data is collected according to the procedures defined in the measurement plan and the feedback material is prepared as described by the analysis plan.

*Step 5:* Feedback sessions. Feedback sessions are meetings involving members of the project team and the measurement team. It is an essential tool for analysis and interpretation of the measurement results A more detailed description of Feedback session is provided hereafter.

*Step 6:* Packaging of measurement results. To re-use measurement results and experiences, the results of the measurement programmes must be packaged. Packaging must be done in such a way that future projects, or other parties from the organisation, are able to use the measurement results.

Application of GQM measurement by means of the described process divides the improvement cycle in two parts. The first part consists of the definition process, during which goals, questions, metrics and additional procedures are defined (step 1, 2, and 3). The second part consists of the interpretation process, during which the collected data is being analysed, improvements are identified, and experiences are described (step 4, 5, and 6). For motivating and convincing software engineers the interpretation process, and especially the feedback sessions are the most important step.

## 3. Interpretation and feedback session

### *3.1 Interpretation process*

Based on the gathered data, an analysis can be performed aimed at answering questions, and reaching goals. This is the “interpretation process”. Research has shown that interpreting measurement data is a learning process within a software team. Interpreting measurement data in a solid way can be done according to the three principles of goal-oriented measurement:

- Software measurement must reflect the interest of the data providers and must be based on the knowledge of the development team.
- Only the software developers that provide the data, can interpret that data validity. They are the only ones who know all the details, also the ones that were *not* measured, and therefore are the only ones allowed to really interpret feedback material.
- Because of the limited amount of time of software developers (caused by their commitments to project planning), conflicts of interest may occur when the development team performs all measurement tasks. Therefore separate staffing must be available that support the collection and analysis of the measurement data, by performing all activities that do not necessarily have to be carried out by the development team. In the context of inspections these activities should be carried out by the inspection implementation team.

To get the most out of measurement, the interpretation must be emphasised to close the feedback loop. The main objective of software measurement is to get interpretations from the project team on the collected data. This data can be used to evaluate current processes and identify improvement opportunities. Depending on the results immediate changes and adjustments on both the software development process and the measurement process can be suggested. Through defining conclusions and action points during the interpretation process, software processes improvement is started at the project and engineering level. The motivation for software developers to participate in software measurement is mainly determined by the way the interpretation process is carried out. The most critical part of the interpretation process are feedback sessions.

### *3.2 Feedback sessions*

Feedback sessions are meetings during which data are analysed by the development team based on the procedures defined in the GQM plan. These sessions are important to keep interest and motivation for the measurement programme. This is one of the reasons why they should be done often. On the other hand, there should be enough time between feedback sessions to ensure there is enough new measurement data and the effort spent is being optimised. This is a kind of paradox: on the one hand, feedback must be done often; on the other hand, this is not feasible. In practice, a balance is achieved by running feedback sessions every six to eight weeks, depending on the project specific goals. Generally feedback sessions last between two and three hours. A feedback session needs a high degree of concentration of the attendees. As an effect, a maximum number of 15 slides presenting measurement results can be discussed. Decisions have to made on the issues to discuss in a feedback session. In the first sessions of a measurement programme, it might be necessary to discuss all available material.

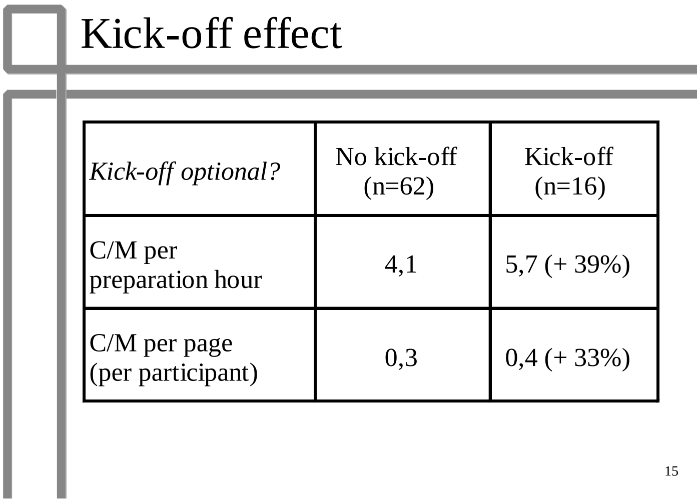
During a inspection feedback session also a changes to the inspection process can be given, new rules or checklists can be introduced. It may also be used to do a survey on the engineers' opinion regarding some aspect of inspection. At one organization feedback session were used to get quantitative data on the logging meeting by asking participant to score statements such as “In the logging meeting we learn how to specify” and “In the logging meeting a common understanding reached”.

| | *Step:* |
|---|---|
| 1. | Freeze Database with measurement data |
| 2. | Create Basic-set of analysis slides |

| | |
|---|---|
| 3. | Create Additional-set with new analysis |
| 4. | Select Feedback-material |
| 5. | Distribute Feedback-material |
| 6. | Analyse Feedback-material in the session |
| 7. | Draw conclusions, answer questions, and define action points |
| 8. | Evaluate with project team |
| 9. | Report |

*Figure 3 : 10 Steps for Feedback Sessions*

Feedback sessions consist of the steps shown in figure 3. The first step is to freeze the database in which the collected measurements are stored. With this data-set it is possible to update the basic set of slides to discuss during the feedback session. Extra slides can also be created to study the specific issues raised by the project team. For instance feedback was asked on average preparation, rework and throughput time. This data wasn't present in the initial set of slides, but was requested by project members to improve their planning. Often the total set of slides is by then already too large. Therefore step 4 must be carried out, during which a selection is made on the subjects for the feedback session. This selection is done in co-operation with the representatives from the project. The slides are always preferably distributed to the attendees of a feedback session one or two days in advance, to offer the opportunity for preparation. Figure 4 shows an example of a slides from an inspection feedback session.



Kick-off effect

| *Kick-off optional?* | No kick-off (n=62) | Kick-off (n=16) |
|---|---|---|
| C/M per preparation hour | 4,1 | 5,7 (+ 39%) |
| C/M per page (per participant) | 0,3 | 0,4 (+ 33%) |

15

*Figure 4 : Example of Inspection feedback session slide*

After the individual preparations, the feedback session can take place. During a feedback session, a representative from the inspection implementation team guides the discussion.. He/she explains which data is included in a presentation slide, explains the axes of a chart, and if a relationship is visible from a slide, points at that relationship. After this explanation, the development team is asked to interpret. Mostly the first interpretation already results in a group discussion, which is finished with an overall interpretation. At the end of the sessions a discussion takes place to identify two or three concrete inspection improvement points. The project team makes a group decision on the improvements to be made. The conclusions are recorded, and the action points are assigned among the participants of the feedback session. A feedback session is finalised by an evaluation with all participants involved. The results, conclusions and inspections improvement actions are reported by the inspection implementation team, including possible improvements for the measurement programme. The feedback process then comes to its end, but since it is a continuous process, measurement on inspection improvement actions are carried out and the next feedback session is already planned……………

## References

[1] Fagan, M.E. (1986), Advances in software inspections, in: IEEE Transactions on Software Engineering, vol. 12, no. 7, July 1986

[2] Gilb, T and D. Graham (1993), Software Inspections, Addison Wesley

[3] Humprey, W.S. (1995), A discipline for software engineering, Addison Wesley

[4] Rooijmans, J., H. Aerts and M. van Genuchten (1996), Software Quality in Consumer Electronic Products, in: IEEE Software, January 1996

[5] Solingen, R. van, and E.P.W.M. van Veenendaal (1997), Achieving software quality by GQM measurement, in: E. van Veenendaal and J. McMullan (eds.), Achieving Software Product Quality, Tutein Nolthenius, ‘s Hertogenbosch, The Netherlands

[6] Solingen, R. van, and E. Berghout (1999), The Goal Question Metrics method, McGrawHill

[7] Van Veenendaal, E.P.W.M (1999), Practical Quality Assurance for Embedded Software, in: Software Quality Professional, Vol. 1, no. 3, June 1999

### The Authors

Erik van Veenendaal
Improve Quality Services BV
Eindhoven University of Technology
The Netherlands
e-mail : eve@improveqs.nl

Mark van der Zwan
Improve Quality Services BV
The Netherlands
e-mail : mzw@improveqs.nl