Using a Process Measurement Framework[SM] to Rapidly Achieve Measurable Results

Abstract

This paper will describe a Process Measurement Framework[SM] that can help organizations to rapidly achieve measurable results. The Process Measurement Framework[SM] is based upon the popular Goal/Question/Metric (G/Q/M) paradigm, the Juran Quality Trilogy, and the initial core measures recommended by the SEI. The G/Q/M Paradigm is applied to the goals of planning, control, and improvement and based on powerful metrics that have a proven track record. In order to illustrate the power of the Process Measurement Framework[SM], real examples from industry are used. Finally, the Process Measurement Framework[SM] helps to ensure that all metrics are collected on a form, in a document, or in a database.

Objectives of this Paper

The objectives of this paper are to:

1. Briefly describe the Goal/Question/Metric Paradigm, the Juran Quality Trilogy, and the SEI recommended initial core measures.
2. Based on objective 1 above, describe a Process Measurement Framework[SM] by providing real examples implemented in industry.
3. Provide some lessons learned using the Process Measurement Framework[SM].

Keywords

Some of the keywords used in this paper are:

- control, database, form, Goal/Question/Metric (G/Q/M), framework, improvement, Juran Trilogy, measure, measurement, metric, planning, process, quality, SEI.

This Paper is Information Mapped

This paper is Information Mapped for non-linear reading. For example, if you already know about the Goal/Question/Metric Paradigm, just skip that section and read the section(s) that you are interested in.

Author

This paper was written by Mr. Timothy G. Olson of Quality Improvement Consultants, Inc. (QIC). Mr. Olson can be reached at (760) 804-1405, and his email is Tim.Olson@qic-inc.com.

In This Paper

This paper contains the following sections:

Goal/Question/Metric Paradigm Overview

Purpose

The purpose of this section is to provide a brief overview of the Goal/Question/Metric Paradigm (see reference below on this page).

G/Q/M Paradigm Summary

The table below provides a high-level summary of each part of the Goal/Question/Metric paradigm:

Part	Description
Goal	Every metric must be directed towards a measurable goal. The idea here is that there must be a good reason to be collecting the data.
Question	Every goal should be answered by one or more key questions. The question should be stated so that a metric can clearly answer it.
Metric	The metric must be a quantitative entity that answers a specific question, which in turn addresses a goal or part of a goal.

The Six Steps of G/Q/M

The table below provides a high-level summary of each step of the Goal/Question/Metric paradigm (from page 729 of the reference below):

Step	Description
1	Establish the goals of the data collection.
2	Develop a list of questions of interest.
3	Establish data categories.
4	Design and test data collection form.
5	Collect and validate data.
6	Analyze data.

Some Problems using G/Q/M

Some common problems using G/Q/M are that users:

- struggle with establishing meaningful goals tailored to their organization.
- have problems coming up with good questions that satisfy those goals.
- struggle with tailoring metrics to the culture of their organization.

Although the Process Measurement Framework[SM] in this paper doesn't completely solve all of these problems, it helps by providing specific goals, questions, and metrics.

Reference

V. R. Basili and D. M. Weiss, "A Methodology for Collecting Valid Software Engineering Data", IEEE Transactions on Software Engineering, vol. SE-10, no. 3, November 1984, pp. 728-738.

Juran Trilogy Overview

Summary of the Juran Trilogy

A financial analogy helps to understand the Juran Trilogy. Managing quality uses the same three fundamental processes as managing finance:

Managing Quality	Analogy: Managing Finance
Quality Planning	Budgeting
Quality Control	Cost Control; Expense Control
Quality Improvement	Cost Reduction; Profit Improvement

Definitions of the Juran Trilogy

Collectively, quality planning, quality control, and quality improvement are called the Juran Trilogy for Quality Management (see reference below). The parts of the Juran Trilogy are defined in the table below:

Part	Definition
Quality Planning	Determining customer needs and developing processes and products required to meet and exceed those customer needs.
Quality Control	Measuring and comparing actual performance against planned performance (e.g., plans, goals, etc.), and taking corrective action on the differences.
Quality Improvement	Eliminating waste, defects, and rework that improves processes and reduces the cost of poor quality.

Tailoring of the Juran Trilogy

The Juran Trilogy has been tailored to the Process Measurement Framework[SM] in the following way:

Part	Definition
Planning	Broadened to be planning in general (e.g., project planning, process planning, product planning, etc.)
Control	Same as the Juran definition above.
Improvement	Broadened to be improvement in general (improving effectiveness, productivity, performance, reducing rework, etc.)

Reference

Juran, Joseph. "The Juran Trilogy", Quality Progress, vol. 19, no. 8, Aug. 1986, pg. 19-24.

SEI Recommended Measures Overview

Purpose

The purpose of this section is to summarize the SEI recommended initial core measures (please see the reference below on this page).

Recommended Measures

The following table describes a summary of the SEI recommended initial core measures (from page 9 of the reference below):

Unit of Measure	Characteristics Addressed
Counts of physical source lines of code	Size, progress, reuse
Counts of staff hours expended	Effort, cost, resource allocations
Calendar dates	Schedule
Counts of software problems and defects	Quality, readiness for delivery, improvement trends

Tailoring of the SEI Measures

The Process Measurement Framework[SM] in this paper tailors the basic SEI measures and adds cost as a separate measurement. Please see the tailored measurements and examples in the table below:

Tailored Measurements	Examples Units of Measure
Cost	Average cost per staff or engineering hour
Defects	Counts of software problems and defects
Effort	Counts of staff hours expended
Schedule	Calendar dates tied to work breakdown structure elements
Size	• Counts of pages of software documentation • Counts of physical source lines of code • Function Points • KSLOC (1000 Source Lines of Code) • KDSI (1000 Delivered Source Instructions)

Summary

The Process Measurement Framework[SM] in this paper uses the basic measurements of cost, defects, effort, schedule, and size in a general sense. These five basic measurements are so powerful, that defect density, performance indexes, and productivity metrics can be derived from them if planned properly.

Reference

"Software Measurement for DoD Systems: Recommendations for Initial Core Measures", By Carleton, Anita D., et al., CMU/SEI-92-TR-19.

A Real Measurement Example: Software Inspections

Purpose

The purpose of this section is to combine the strengths of the G/Q/M paradigm, the Juran Trilogy, and the SEI basic measurements, and tailor them into a powerful Process Measurement Framework[SM].

Form Acronyms

The acronyms used for inspection forms for data collection are:

MN = Meeting Notice

DL = Defect List

DS = Defect Summary

MS = Meeting Summary

Example: Process Measurement Framework[SM] for Planning Inspections

The G/Q/M Paradigm is used across the top of the matrix or Framework (i.e., column headings). The Juran Trilogy of plan, control, and improve are used as row headings (over the next three pages). The Framework is filled in using the tailored SEI recommended measurements of cost, defects, effort, schedule, and size. The Framework below has been designed for inspections, and describes the goals, key questions for each goal, metrics that answer those questions, and which form the data is collected on:

Goal	Key Questions	Basic Metrics	Data Collection
1. Plan and Estimate within 10% of Actuals • **Use historical data**	Per Work Product: 1a). How much will the inspections cost? How much will defects cost?	Based on Work Product Size: 1a). Average **cost** per page (e.g., $50.00 per page). Average cost per defect (e.g., $100 per defect).	Inspection Database
	1b). How many defects will there be?	1b). **Defect** density (e.g., average 0.5 defects per page).	Inspection Database
	1c). How much effort will the inspection take? per defect?	1c). Average **effort** per page (e.g., 1 hour per page). Average effort per defect (e.g., 2 hours per defect).	Inspection Database
	1d). How long will the inspections take?	1d). Inspection **Schedule** (based on average preparation rate and average meeting rate and 2 hour limit duration per meeting)	Software Project Plan
	1e). How big is the work product?	1e). Total work product **size** in pages (e.g., 300 page design document).	Inspection Database

A Real Example: Software Inspections, Continued

A Real Example

Based on the example Process Measurement Framework[SM] for planning on the previous page, to inspect a design document of 300 pages is size could be:

Inspection Measurements	Example Estimates
Total Size	100 pages (Design Document)
Total Defects	100 Total Defects (100 pages * 1 defect per page)
Total Cost	• $5,000 to inspect document (100 pages * $50 a page) • $50 average cost per defect ($5,000/100 defects)
Total Effort	• 100 person hours of effort (1 hour per page * 100 pages) • 1 hour average effort per defect (100 hours/100 defects)
Schedule	• Average preparation rate of 10 pages per hour = 10 hours • Average meeting rate of 10 pages per hour = 10 hours • 10 hours/2 hour meetings approximately 5 meetings • Schedule = 2-4 weeks (calendar time) for entire process

Example: Process Measurement Framework[SM] for Control

The definition of control according to Dr. Juran is "comparing actual measurements against planned measurements, and taking action on the difference." The table below describes the control goal, the key questions, metrics that answer those questions, and which form the data is collected on.

Goal	Key Questions	Basic Metrics	Data Collection

2. Control • **Measure and track actual data against estimated data** • **Take action on major differences (Greater than 10%)**	What is the inspection status (per work product)? 2a) What do the inspections cost? per defect? 2b) How many defects are there? What is the quality? 2c) How much effort do the inspections take? per defect? 2d)What is the schedule status? 2e) How many pages have been inspected?	Measure actual data against estimated data: 2a) Actual average **cost** per page vs. estimated. Actual average **cost** per defect vs. estimated. 2b) Total number of **defects**. Actual **defect** density vs. estimated. 2c). Actual average **effort** per page vs. estimated. Actual average **effort** per defect vs. estimated. 2d) **Schedule**: Percentage of actual inspections complete vs. estimated). 2e) **Size**: Total actual pages inspected to date vs. estimated.	Note: All metrics in inspection database Derived MS MS MS MS

A Real Example: Software Inspections, Continued

Example: Process Measurement Framework[SM] for Improving Inspections

The definition of improvement according to Dr. Juran is eliminating waste, defects, and rework that improves processes and reduces the cost of poor quality. The table below describes the improvement goal, the key questions, metrics that answer those questions, and which form the data is collected on:

Goal	Key Questions	Basic Metrics	Data Collection
3. Improve Performance **• Improve inspection process based on data**	Per work product, what is the inspection performance of? 3a) cost? 3b) defect density? 3c) effort? 3d) schedules? 3e) work product size?	Improve performance indexes to become close to 1.0 (actual data over estimates): 3a) **Cost** performance index 3b) **Defect** performance index 3c) **Effort** performance index 3d) **Schedule** performance index 3e) **Size** Performance index	Actual Data for all questions: Inspection Database Data Analysis Tools
4. Improve Inspection Effectiveness **• Improve inspection process based on data**	4a) How effective is the inspection process? 4b) What defects did the inspections miss in the testing phase(s)? 4c) What are the vital few defect categories that cause 80% of all defects? 4d) What is the 20% of the code that causes 80% of the defects?	4a) **Defect**-removal efficiency (i.e., percentage of all **defects** found by inspections during the entire process for a given work product) 4a) Average **cost** and **effort** per **defect** 4b) **Defects** in test and/or SCM databases 4c) Pareto analysis of total **defects** in **defect** categories (per work product, by phase, etc.) 4d) **Defect** location (from SIDL form. Advanced: also related to software complexity measures).	Inspection Database Test Database SCM Database All defect databases Data Analysis Tools

5. Optimize Inspection Process • **Improve inspection process based on data**	5a) What is the optimum effectiveness and productivity of the inspection process?	5a) Measure the relationships among metrics (e.g., using scatter diagrams): Work product **size** Average **cost** per **defect** **Defect** Density Average **effort** per **defect** Average **effort** per **pages** Average preparation rate Average inspection rate Average **pages** inspected	Actual Data for all questions: Inspection Database Data Analysis Tools

Summary: Lessons Learned and References

Purpose

The purpose of this section is to describe some lessons learned when using the Process Measurement Framework[SM], and to provide some references from the literature.

Some Lessons Learned

Some of the lessons learned while using the Process Measurement Framework[SM] are:

- The Process Measurement Framework[SM] is very powerful once you get used to it. Using a good inspection process and the Process Measurement Framework[SM] will average 7:1 ROI and many other measurable results.
- Writing "good questions" in the G/Q/M paradigm is challenging. The Process Measurement Framework[SM] adds more structure to the goals (e.g., the Juran Trilogy) and to the metrics (e.g., the SEI recommended measurements), which helps to make the questions easier to write.
- Operational definitions for each metric are required for repeatability.
- There are many metrics that are derivable from the basic five metrics. For example, productivity (effort/size), defect density (defects/size), and performance (estimates/actuals).
- The Process Measurement Framework[SM] needs to be based on a measurement process (e.g., six steps in G/Q/M).
- There are other goals such as prevention and return on investment (ROI) the Process Measurement Framework[SM] can help implement.
- The Process Measurement Framework[SM] must be tailored to each organization, division, and even to each project.

References

The references used for this paper are:

- [Barnard 1994] Barnard, J. and Price, A. "Managing Code Inspection Information", IEEE Software, March 1994.
- [Basili 1984] Basili, V. R., and Weiss, D. M., "A Methodology for Collecting Valid Software Engineering Data", IEEE Transactions on Software Engineering, vol. SE-10, no. 3, November 1984, pp. 728-738.
- [Carleton 1994] Carleton, Anita D., et al., "Software Measurement for DoD Systems: Recommendations for Initial Core Measures", CMU/SEI-92-TR-19.
- [Ebenau 1994] Ebenau, B. and Strauss, S., *Software Inspection Process*. McGraw-Hill, 1994.
- [Fagan 1986] Fagan, M. "Advances in Software Inspections", M. IEEE Transactions on Software Engineering, July 1986
- [Grady 1994] Grady, R. and Van Slack, T. "Key Lessons In Achieving Widespread Inspection Use", IEEE Software, July 1994.
- [Herbsleb 1994] Herbsleb, James, et al, "Benefits of CMM-Based Software Process Improvement: Initial Results", CMU/SEI-94-TR-13, 1994.
- [Juran 1989] Juran, Joseph. *Juran on Leadership for Quality: An Executive Handbook*. New York, NY: Macmillan, 1989.
- [Olson 1995] Olson, Timothy G. "A Software Quality Strategy for Demonstrating Early ROI", Society of Software Quality Journal, May 1995.
- [Olson 1997] Olson, Timothy G., "A Measurement Framework That Works", Presentation, SEPG 1997, San Jose, CA.

• [Putnam 1992] Putnam, L., and Myers, W. *Measures for Excellence: Reliable Software On Time Within Budget.* Prentice-Hall, 1992.

