QUANTITATIVE ASSESSMENT OF MAINTENANCE:
An Industrial Case Study

H. Dieter Rombach and Victor R. Basili

University of Maryland
Dept. of Computer Science
College Park, MD 20742, USA
(301) 454-8974

Abstract

In this paper we discuss a study aiming at the improvement of measurement and evaluation procedures used in an industrial maintenance environment. We used a general evaluation and improvement methodology for deriving a set of metrics tailored to the maintenance problems in this particular environment. Some of the required maintenance data were already collected in this environment, others were suggested to be collected in the future. We discuss the general measurement, evaluation and improvement methodology used, the specific maintenance improvement goals important to this environment, the set of metrics derived for quantifying those goals, the suggested changes to the current data collection procedures, and preliminary analysis results based on a limited set of already available data. It is encouraging that based on this limited set of data we are already able to demonstrate benefits of the proposed quantitative approach to maintenance. Finally, we outline ideas for automating the discussed approach by a set of measurement and evaluation tools. This paper emphasizes the steps of introducing such a quantitative maintenance approach into an industrial setting rather than the environment-specific analysis results. The analysis results are intended to demonstrate the practical applicability and feasibility of the proposed methodology for evaluating and improving maintenance aspects in an industrial environment.

1. Introduction

In this paper we present results from a study trying to introduce sound measurement and evaluation procedures into an industrial maintenance environment. The goal of the study has been to investigate the company's needs for quality assessment, and the suitability of the error, change, and effort data already collected in this environment for addressing these quality assessment needs.

First we describe the actual industrial maintenance environment which has been the object of this study including the high-level maintenance assessment and improvement goals as stated by high-level management (section 2) and the goal/question/metric paradigm[1, 3, 7] used in this study for defining and quantifying the maintenance assessment and improvement goals of interest. The application of this methodology has resulted in a list of clearly defined maintenance assessment and improvement goals and quantifiable questions (section 4) as well as the corresponding data and metrics (section 5). Until now only a subset of these data and metrics required to fully address the stated maintenance goals had been collected (section 6). Based on the needs of the particular industrial environment changes to the data collection and validation process have been suggested for the future (section 7). Preliminary analysis results for a small subset of the questions and goals of interest (depending on the type, amount and quality of data available at the time) are presented (section 8). It is encouraging that based on this limited subset of data we are already able to demonstrate benefits of this quantitative approach to maintenance. Finally, we outline ideas for automating the proposed approach by a set of measurement and evaluation tools (section 9). This paper emphasizes the steps of introducing such a quantitative maintenance approach into an industrial setting rather than the environment-specific analysis results. The analysis results are only included to demonstrate that the proposed approach actually works in this particular environment.

2. Maintenance Environment

The study was conducted in the maintenance environment of a major computer company. The maintenance process from an organizational point of view can be characterized as follows: **Customer Support** receives maintenance problems (mainly) from customers, evaluates them and, whenever appropriate forwards them in the form of change requests to **Product Assurance**. Product Assurance evaluates the change requests again and forwards them, whenever appropriate, to **Engineering**. The eventually changed products are sent back to the customer(s) through the same channels (Product Assurance, Customer Support).

Data are currently being collected during all these different maintenance steps. Customer Support collects data for each single problem concerning scheduling (e.g., time of incoming calls, time of outgoing calls), type of problem (e.g., clarification of documentation, operation request; for a complete list see table 2), priorities of problems, and effort spent on handling the problem. Product Assurance collects data for each single change request concerning scheduling, type of change request, effort spent, and final status (e.g., changed, change postponed, change rejected including the reason for

This study was supported by a grant from Burroughs Corporation to the University of Maryland. Computer time was provided in part through facilities of the Computer Science Center of the University of Maryland.

CH2442-2/87/0000/0134$01.00 © 1987 IEEE

rejection). Engineering collects data for each change concerning scheduling, change effort, and the type of change performed. Data collection is mandatory in some groups such as Product Assurance; it is done on a voluntary basis in other groups such as Engineering. Based on this fact the completeness and validity of collected data varies across the entire maintenance environment. In general it is true that Customer Support and Product Assurance stress data collection more than Engineering does.

Although this is a very simplified description of the maintenance process it should allow the reader to understand the different needs of these three different maintenance roles as far as assessment needs are concerned.

The data were used for filing status reports concerning the handling of maintenance requests but not (except locally in some groups) for overall quality assessment. The purpose of this study was to find out whether the already collected data are sufficient for assessing the environment specific maintenance problems and, if not, to suggest changes of this data collection process.

The most urgent maintenance assessment and improvement goals were formulated by corporate representatives of the company as follows:

G1: Examine where the bulk of the company's maintenance dollars are being spent and how much is being spent on individual activities.

G2: Identify the best ways of applying the 20/80 rule* to get the biggest savings and return on our maintenance dollars.

G3: Identify criteria for when a product is ready for release.

G4: Identify features of product, documentation or support that provide a wider customer satisfaction.

G5: Identify criteria for when a software product should be rewritten rather than maintained.

G6: Identify metrics of customer satisfaction that can be developed based upon existing data.

G7: Develop organizational guidelines for integrating software quality metrics into the company's framework of design, development, and support.

It is obvious that these high-level and complex problems can only be assessed by breaking them down into more and more simple problems. This refinement process, which finally is expected to result in a set of quantitative metrics, is supported by a methodology developed by the authors[1, 3, 7].

3. The Goal/Question/Metric Paradigm

The approach to quantification of goals is the goal/question/metric paradigm[1, 3, 5, 7]. This paradigm does not provide a specific set of goals but rather a framework for defining goals and refining them into specific quantifiable questions about the software process and product that provide a specification for the data needed to help answering the goals.

The paradigm provides a mechanism for tracing the goals of the collection process, i.e. the reasons the data are being collected, to the actual data. It is important to make clear, at least in general terms, the organization's needs and concerns, the focus of the current project and what is expected from it. The formulation of these expectations can go a long way towards focusing the work on the project and evaluating whether the project has met those expectations. The need for information must be quantified whenever possible and the quantification analyzed as to whether or not it satisfies the needs. This quantification of the goals should then be mapped into a set of data that can be collected on the product and the process. The data should then be validated with respect to how accurate it is and then analyzed and the results interpreted with respect to the goals.

The actual goal/question/metric paradigm is visualized in figure 1.

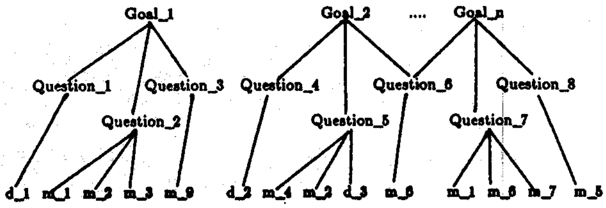

Figure 1: Goal/Question/Metric Paradigm.

Here there are n goals shown and each goal generates a set of questions that attempt to define and quantify the specific goal which is at the root of its goal tree. The goal is only as well defined as the questions that it generates. Each question generates a set of metrics (m_i) or distributions of data (d_i). Again, the questions can only be answered relative to and as completely as the available metrics and distributions allow. As is shown in figure 1, the same questions can be used to define different goals (e.g. Question_6) and metrics and distributions can be used to answer more than one question (e.g. m_1 and m_2). Thus questions and metrics are used in several contexts.

Given the above paradigm, the **process of quantifying improvement goals** consists of three steps:

(1) Generate a set of goals based upon the needs of the organization.
The first step of the process is to determine what it is you want to improve. This focuses the work to be done and allows a framework for determining whether or not you have accomplished what you set out to do. Sample goals might consist of such issues as on how to improve the set of methods and tools to be used in a project with respect to high quality products, customer satisfaction, productivity, usability, or that the product contains the needed functionality.

(2) Derive a set of questions of interest or hypotheses which quantify those goals.
The goals must now be formalized by making them quantifiable. This is the most difficult step in the process because it often requires the interpretation of fuzzy terms like quality or productivity within the context of the development environment. These questions define the goals of step 1. The aim is to satisfy the intuitive notion of the goal as completely and consistently as possible.

* Applying the 20/80 rule means to identify those maintenance problems which can be fixed easily (with twenty percent of the effort of what would be required to fix all maintenance problems) but reduce the maintenance overhead drastically (by eighty percent).

(3) Develop a set of data metrics and distributions which provide the information needed to answer the questions of interest.
In this step, the actual data needed to answer the questions are identified and associated with each of the questions. However, the identification of the data categories is not always so easy. Sometimes new metrics or data distributions must be defined. Other times data items can be defined to answer only part of a question. In this case, the answer to the question must be qualified and interpreted in the context of the missing information. As the data items are identified, thought should be given to how valid the data item will be with respect to accuracy and how well it captures the specific question.

In writing down goals and questions, we must begin by stating the purpose of the improvement process. This purpose will be in the form of a set of overall goals but they should follow a particular format. The format should cover the purpose of the process, the perspective, and any important information about the environment. The format (in terms of a generic template) might look like:

- **Purpose of Study**:
  To (characterize, analyze, evaluate, predict, motivate) the (process, product, model, metric) in order to (understand, assess, manage, engineer, learn, improve) it.
- **Perspective of Study**:
  Examine the (cost, effectiveness, correctness, errors, changes, product metrics, process metrics, reliability, user satisfaction, etc.) from the point of view of the (developer, manager, customer, corporate perspective, etc).
- **Environment of Study**:
  The environment consists of the following: process factors, people factors, problem factors, methods, tools, constraints, etc.
- **Process Questions**:
  For each process under study, there are several subgoals that need to be addressed. These include the quality of use (characterize the process quantitatively and assess how well the process is performed, the domain of use (characterize the object of the process and evaluate the knowledge of object by the performers of the process), effort of use (characterize the effort to perform each of the subactivities of the activity being performed), effect of use (characterize the output of the process and the evaluate the quality of that output), and feedback from use (characterize the major problems with the application of the process so that it can be improved).

  Other subgoals involve the interaction of this process with the other processes and the schedule (from the viewpoint of validation of the process model).
- **Product Questions**
  For each product under study there are several subgoals that need to be addressed. These include the definition of the product (characterize the product quantitatively) and the perspective of the evaluation (e.g. reliability or user satisfaction). The definition of the product includes physical attributes (e.g. source lines, number of units, executable lines, control and data complexity, programming language features, time space), cost (e.g. effort, time, phase, activity, program), changes (e.g. errors, faults, failures and modifications by various classes), and the context the product is supposed to be used in (e.g. customer community, operational profile). The perspective of the evaluation is relative to a particular quality (e.g. reliability or user satisfaction). Thus the physical characteristics need to be analyzed relative to this quality aspect.

4. Maintenance Goals and Questions

We applied the methodology described in section 3 to specify the high-level quality assessment and improvement goals given to us from a corporate perspective (see section 2) more precisely, and to derive quantifiable analysis questions. Using the template of section 3 proved to be very helpful. The entire process of specifying goals and deriving the evaluation questions was done in very close cooperation with company representatives from Customer Support, Product Assurance, and Engineering.

The seven goals for this study are formulated in terms of the purpose of this study, the perspective of this study, and important information about the company's maintenance environment:

- PURPOSE OF STUDY: Characterize (in the case of goals G1 and G4) and evaluate (G2, G3, and G5) the maintenance methodology and motivate (G6 and G7) the use of metrics for the purpose of better understanding (G1 and G4), management (G2, G3, G5, G6, and G7) and improvement (G2, G3, G5, G6, and G7).
- PERSPECTIVE: Examine the cost (in the case of goals G1, G2, G5, and G7), problems (G2), errors and changes (G1 and G5), product and process metrics (G3, G4, G5, and G6) and the effectiveness (G7) from the point of view of the manager and corporation.
- ENVIRONMENT:
  - Maintenance Process: The customer reports problems (by phone) to the Customer Support; if problems cannot be resolved by Customer Support they are forwarded to Product Assurance. Product assurance decides whether the reported problem should be fixed. If approved as a problem to be fixed it is submitted to engineering (to be fixed), gets back to Product Assurance (for fix certification), and is sent back to Customer Support.
  - Maintained Products (for which we had access to data): A retrieval system (called SYS_1 in the following of this paper) and a compiler (called SYS_2 in the following of this paper).

For each process and product under study, there are several subgoals (quality of use, domain of use, effort of use, effect of use, and feedback of use); each subgoal will be addressed by a number of analysis questions (Qi):

(A) PROCESS RELATED QUESTIONS:

- **QUALITY OF USE (characterize the company's maintenance process and how well it is performed):**

  Q1: What percent of the problems are handled by Customer Support without forwarding them to Product Assurance? What is a distribution of their disposition?

Q2: **What percent of change requests forwarded to Product Assurance do not come from the field? What is a distribution by percent of where they come from (engineering, field test, etc) and the reasons they do not come from field? What percent of problems aren't really maintenance problems?**

Q3: For change requests rejected by Product Assurance or Engineering: What are the distributions by

1) closure code,
2) organization responsible for rejection, and
3) schedule by closure code by organization?

Q4: What are characteristics of the test plan performed by engineering before release? How effective is this test plan?
More detailed: Is the test suite based upon the new or changed final requirements? Are regression tests performed? Are the tests based upon the importance and complexity of the requirements? What criteria exist for the selection of test cases and test data?

Q5: What are test cases and test data for the beta test? To what extent does it consider the future usage profile? How effective is this test?

Q6: For each fix: How long after the fix is made is it released to the customer?

Q7: What is the distribution of faults or customer problems per organizational unit in total and by various products?

Q8: What is the distribution of faults due to previous changes per organizational unit in total and by various products?

Q9: What are the distributions of change requests by various subclasses (fault/modification, rejected/not rejected, error subclasses, change subclasses)?

- **DOMAIN OF USE (characterize the objects of the maintenance process and the knowledge of the people involved in this maintenance process):**

Q10: What products are available to

- customer support personnel,
- problem evaluator,
- changer,
- change evaluator, and
- the field support?

Q11: What is the knowledge of the people involved wrt

1) the application,
2) the particular product, and
3) the change methodology?

- **EFFORT OF USE (characterize the effort to perform each maintenance activity):**

Q12: What is the cost of

- detecting a problem symptom
- understanding the problem,
- isolating the problem causes,
- designing the change,
- implementing the change,
- testing the change, and
- releasing the change

in terms of computer time, people time, by person category and machine category?

Q13: What is the calendar time for

- detecting a problem symptom,
- understanding the problem from a customer's viewpoint,
- understanding the problem from an engineering viewpoint,
- isolating the problem causes,
- designing the change,
- implementing the change,
- testing the change, and
- releasing the change?

[Give the max, min, average and by various types of changes!]

- **EFFECT OF USE (characterize the output of the maintenance process and the quality of this output):**

Q14: How many and what percent of documents are produced/modified as a result of the maintenance process (patch, user manual, additional technical documents, closure form, patch release information form, advanced technical information form and user letter)?

Q15: How many and what percent of change requests cause a modification?

Q16: How many and what percent of change requests are related to errors, environment adaptations, and requirements changes (= enhancements)?

Q17: How many and what percent of faults are the result of a previous change?

Q18: What is the average cost of a change overall and by type?

Q19: Having categorized changes by function, having made a change in a function: How many future requests do we get for the same function?

Q20: What are characteristics of customer calls over time by type of question?

Q21: What customer categories exist? Do clusters of customer profiles (types of complaints, faults, etc.) match these categorization schemes?

Q22: Is the user satisfied with function, performance, schedule (by a user satisfaction survey)?

- **FEEDBACK FROM USE (characterize the problems with the application of the maintenance process so that it can be improved):**

Q23: What are the problem areas in the maintenance process by the following categories:

- distribution of changes by various types,
- distribution of problems that are rejected by various types,
- customer types, and
- time distribution (calendar time, effort) by various change types, problem types, or maintenance activities?

(B) PRODUCT RELATED QUESTIONS:

- **DEFINITION OF THE PRODUCT (characterize the product quantitatively):**

Q24: What are the physical attributes such as

- size (source lines, number of units, executable lines of code),
- complexity (control, data),
- programming language features,
- time to develop,
- memory space, and
- execution frequency?

Q25: What is the cost, e.g., effort (time per phase, activity)?

Q26: What are distributions of changes, e.g., errors, faults, failures, adaptations, and enhancements by various types

GOALS	QUESTIONS																											
	PROCESS																							PRODUCT				
	QUALITY OF USE									DOMAIN OF USE		EFFORT OF USE		EFFECT OF USE														
	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10	Q11	Q12	Q13	Q14	Q15	Q16	Q17	Q18	Q19	Q20	Q21	Q22	Q23	Q24	Q25	Q26	Q27	Q28
G1	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
G2	x	x	x	x	x		x	x	x	x	x	x	x	x	x	x	x	x	x	x	x		x		x	x	x	x
G3		x		x	x		x	x	x		x										x			x		x	x	x
G4	x		x	x	x	x	x	x	x	x	x		x	x		x	x		x	x	x	x		x		x	x	x
G5					x		x	x	x	x	x	x					x		x					x	x	x		x
G6	x		x			x	x	x	x	x			x	x		x	x		x	x				x		x		x
G7	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

Table 1: Goal-Question Matrix

Q27: What is the products context, e.g, customer community, operational profile, life cycle model, etc?

Q28: What are the problem areas in the product by the following categories:

- distribution of changes by various types,
- distribution of problems that are rejected by various types,
- customer types, and
- time distribution (calendar time, effort) by various change types, problem types, or maintenance activities?

Each individual evaluation goal is quantifiable via a subset of these 28 evaluation questions. In table 1 the interrelationship is visualized in form of a goal-question matrix.

5. Maintenance Data & Metrics

In this section we discuss the types of maintenance data which has to be collected in order to answer each of the evaluation questions derived in section 4.

The data (Di) are categorized depending on which maintenance aspect (Customer Support, Product Assurance, or Engineering) is mainly affected. For each data it is indicated whether and how it can be retrieved from currently maintained data bases, i.e., whether it is explicitly available (+), it is not explicitly available, but can be derived from other data with reasonable effort (o), a great deal of effort (oo), or it is not available at all (-).

(1) CUSTOMER SUPPORT ORIENTED MAINTENANCE DATA:

For each problem reported by customers (phone calls):

D1 (+): customer identification
D2 (oo): customer type
D3 (+): customer support center identification
D4 (o): problem description
D5 (+): whether a problem resulted in a change request (Y/N)
D6 (oo): connection between customer problem and change request_number
D7 (+): identification of affected system/product
D8 (-): identification of affected product functions
D9 (+): schedules for each activity associated with a customer problem

(2) PRODUCT ASSURANCE ORIENTED MAINTENANCE DATA:

For each problem reported by a change request:

D10 (+): identification of the organization that filled out the change request (customer support, engineering, field test, etc)
D11 (+): identification of system/product affected
D12 (+): customer identification
D13 (-): customer type
D14 (+): identification of Product Assurance center in charge
D15 (o): concise problem description
D16 (o): information whether a change request was rejected (Y/N)
D17 (+): final change request status (= closure code)
D18 (-): information by whom (Product Assurance, Engineering) closure code was set
D19 (+): schedules for each maintenance activity
D20 (+): information whether it is a fault, adaptation, or enhancement

(3) ENGINEERING ORIENTED MAINTENANCE DATA:

For each actually performed change:

D21 (+): identification of the engineering group in charge
D22 (-): information about fault types (for example: control, data, computation, etc)
D23 (o): information whether a fault was caused by a previous change (Y/N)
D24 (o): information which product units (modules) were affected by a change (in terms of lines_of_code or identification of modules)
D25 (-): effort in computer time in total or per phase, change activity
D26 (-): effort in people time in total or per phase, change activity
D27 (+): schedule for each change activity (in calendar days)
D28 (o): percent of code, documents, forms changed
D29 (o): product size
D30 (o): product complexity
D31 (-): memory space

The following question-data matrix (see table 2) shows which of the 31 different types of data are required as a minimum to answer each of the previously listed 28 questions:

QUESTIONS	DATA/MEASURES																														
	D1	D2	D3	D4	D5	D6	D7	D8	D9	D10	D11	D12	D13	D14	D15	D16	D17	D18	D19	D20	D21	D22	D23	D24	D25	D26	D27	D28	D29	D30	D31
Q1				x	x			x																							
Q2										x																					
Q3														x		x	x	x													
(Q4)																															
(Q5)																															
Q6																			x												
Q7			x	x			x				x			x	x		x			x	x	x									
Q8			x	x			x				x			x	x		x			x	x		x								
Q9																x	x			x		x									
(Q10)																															
(Q11)																															
Q12																									x	x					
Q13				x					x																		x				
Q14																												x			
Q15																	x														
Q16																	x			x											
Q17																				x			x								
Q18																	x					x			x	x					
Q19					x	x		x																							
Q20				x																											
Q21	x	x		x								x	x				x														
(Q22)																															
Q23																															
Q24																													x	x	x
Q25																									x	x					
Q26															x		x			x		x		x							
(Q27)																															
Q28																															

Table 2: Question-Data Matrix

The questions enclosed in parenthesis have to be answered purely by subjective data.

The complete refinement process from the original goals over questions to the data/metrics can be traced by combining tables 1 and 2.

6. Availability and Validity of Data

In the previous section it was indicated what data are needed for answering the questions of interest. We also included the analysis results to which degree those data are already available inside the company (+,o,-).

Interpreting the question/data matrix together with the availability and validity of the company's data the following conclusions can be drawn:

- Questions Q6, Q13, Q15, Q16, Q17, Q20 are completely answerable
- Questions (Q4), (Q5), (Q10), (Q11), (Q22) will not be answered based on data collected via regular data collection forms, but by subjective data from interviews.
- Questions Q23 and Q28 require no data, they are answered by interpreting the results of more basic questions
- All questions related to change effort (Q12, Q18, Q25) can not be answered because (at least in the case of SYS_1 and SYS_2) these data were listed as optional on the data collection form and therefore only listed on about 10% of all forms.
- All other questions are (at least partially) answerable

7. Improvement of Data Collection

Based on the company's interests as documented by the high-level problems (see section 2) and the refined set of evaluation questions (see section 4), and the partial lack of valid data available to analyze those questions, the following recommendations for changing the data collection process are being made:

- A uniform data collection method and data base should be defined..
  Some data items are interpreted differently by different people. Each organizational unit inside the the maintenance environment has its own data base format. This fact makes it difficult to assess maintenance problems from global views. It is for example difficult to analyze engineering data from various sites, or the complete life cycle of maintenance problems starting at Customer Support throughout Product Assurance and Engineering.
- A maintenance task should be viewed as a single entity in this data base, and it should be traceable through all its phases (Customer Support, Product Assurance, Engineering). Due to the "bottom-up" development of individual data bases, each data base contains only those data important to the individual organization.
  The only solution seems to be a central data base that contains all information concerning each maintenance task starting from the first phone call and ending with its final resolution.
- It is mandatory to collect engineering data (effort in staff_hours).
  Engineering data are crucial for determining maintenance problems due to product quality problems (e.g., bad structure).
- Development data (errors, changes, tests, etc.) should be collected.
  Collection of development data has to start now. As soon as the identification of the maintenance problems is completed, the impact of product quality and development methodology on these problems has to be analyzed. In order to do this, data characterizing the development process are needed.

8. Preliminary Analysis Results

In order to demonstrate the benefits of quantitative assessment of maintenance we used the data collected at the time to answer some of our maintenance questions listed in section 4. We had data available for two commercial systems SYS_1 and SYS_2 (retrieval system and a compiler). We had maintenance data available from the first two quarters of 198o.

In section 6 we outlined the questions which could be answered based on the data available. In the following we present preliminary analysis results of those questions in the context of the originally posted high-level corporate maintenance problems (1) to (5) as listed in section 2.

(G1) Examine where the bulk of the company's maintenance dollars are being spent and how much is being spent on individual activities:

This goal area can be addressed by the following analysis questions (see section 4):

- Question 20: (What are characteristics of customer calls over time by type of question ?) ---> Table 3

  The average number of calls per problem is about 4. The most frequent problems are operation questions, capability features, and clarification of documentation (in the case of SYS_1) or operation fault (in the case of SYS_2). The costly problems (in terms of number of calls) are documentation faults, system software, and operation faults (in the case of SYS_1), and clarification of documentation, capability features, operation questions, and pre-sales requests (in the case of SYS_2).

- Question 1 (What percent of problems are not reported as change requests? What is a distribution of their disposition?) ---> Table 4

  Overall only about two percent of all problems recorded by Customer Support resulted in change requests (3 out of 177 for SYS_1, 3 out of 152 for SYS_2).

  The disposition of problems not reported as change requests in terms of "type of call" is as follows:

  The bulk of maintenance problems haadled by Customer Support is spent for "operation requests" and "operation faults" in the case of SYS_2; in the case of SYS_1 we can identify two additional problem sources: problems due to faults of underlying layers (systems software and hardware) and problems due to bad documentation (almost 20% of all problems !)

call-type	SYS_1			SYS_2		
	calls	problems	calls/problem	calls	problems	calls/problem
unknown type	5	2 (1.1%)	2.5	-	-	-
clarify document	130	35 (19.8%)	3.7	34	7 (4.6%)	4.9
operation question	172	46 (26.0%)	3.7	378	78 (51.3%)	4.8
pre-sales request	7	2 (1.1%)	3.5	9	2 (1.3%)	4.5
capability, feature	88	30 (16.9%)	2.9	84	17 (11.2%)	4.9
other	43	13 (7.3%)	3.3	61	19 (12.5%)	3.2
document fault	7	1 (0.6%)	7.0	-	-	-
operation fault	50	10 (5.6%)	5.0	44	20 (13.2%)	2.2
application SW change request	-	-	-	3	1 (0.7%)	3.0
application SW fault	4	1 (0.6%)	4.0	-	-	-
system SW fault	85	15 (8.5%)	5.7	15	4 (2.6%)	3.8
system SW change request	14	3 (1.7%)	4.7	6	2 (1.3%)	3.0
instruction fault	7	2 (1.1%)	3.5	-	-	-
HW fault	67	17 (9.6%)	3.9	5	2 (1.3%)	2.5
AVERAGE			3.7			4.1

Table 3: All Calls/Problems by Call-Type

call-type	SYS_1			SYS_2		
	calls	problems	calls/problem	calls	problems	calls/problem
unknown type	5	2 (1.1%)	2.5	-	-	-
clarify document	130	35 (19.8%)	3.7	34	7 (4.6%)	4.9
operation question	172	46 (26.0%)	3.7	378	78 (51.3%)	4.8
pre-sales request	7	2 (1.1%)	3.5	9	2 (1.3%)	4.5
capability, feature	88	30 (16.9%)	2.9	84	17 (11.2%)	4.9
other	43	13 (7.3%)	3.3	61	19 (12.5%)	3.2
document fault	7	1 (0.6%)	7.0	-	-	-
operation fault	50	10 (5.6%)	5.0	44	20 (13.2%)	2.2
application SW fault	4	1 (0.6%)	4.0	-	-	-
system SW fault	85	15 (8.5%)	5.7	15	4 (2.6%)	3.8
instruction fault	7	2 (1.1%)	3.5	-	-	-
HW fault	67	17 (9.6%)	3.9	5	2 (1.3%)	2.5
TOTAL	665	174/177 (98.3 %)	3.7	630	149/152 (98 %)	4.1

Table 4: Non-forwarded Calls/Problems by Call-Type

- Question 2 (What percent of problems aren't really maintenance problems?) ---> Table 5

Table 5: Portion of Real Maintenance Problems

	SYS_1	SYS_2
Number of total problems	177	152
Number of maintenance problems	80	116
percentage	45.2 %	76.3 %

Not all of the problems reported to Customer Support are really maintenance problems. There are, for example, lots of requests from different divisions inside the company. From a global view, all the effort spent in Customer Support is charged as maintenance effort. In the case of SYS_1, only about 45% of all problems (80 out of 177), and in the case of SYS_2, only about 76% of all problems (116 out of 152) are really maintenance problems.

- Question 3 (What is the distribution of rejected change requests by closure code?) ---> Table 6

  The distribution of rejected change requests by closure code is as follows:

Table 6: Rejected Change Requests by Closure Code

Closure Code	Systems	
	SYS_1	SYS_2
need additional information	11	11
not reproducible	1	-
no fix scheduled	3	2
already fixed	45	25
forwarded to ...	-	2
works as intended	6	1
works as documented	-	3
incorrect documentation	2	-
operation problem	1	1
document required	1	-
not retrofit	2	8
other	-	2

- Question 12 (What is the cost of ......?)

  Because we have no effort data concerning the Product Assurance and engineering aspects of the maintenance process, we only could analyze effort as far as Customer Support was concerned:

  The cost for each individual maintenance problem (as far as Customer Support is concerned) can be characterized

call-type	SYS_1			SYS_2		
	time (mins)	problems	time/problem	time (mins)	problems	time/problem
unknown type	52	2	26.0	-	-	-
clarify document	791	35	22.6	247	7	35.3
operation question	1203	46	26.1	3723	78	47.7
pre-sales request	36	2	18.0	211	2	105.5
capability, feature	739	30	24.6	747	17	44.0
other	247	13	19.0	813	19	42.8
document fault	43	1	43.0	-	-	-
operation fault	303	10	30.3	522	20	26.1
application SW change request	-	-	-	20	1	20.0
application SW fault	53	1	53.0	-	-	-
system SW fault	509	15	33.9	78	4	19.4
system SW change request	167	3	55.8	8	2	4.0
instruction fault	13	2	6.6	-	-	-
HW fault	327	17	19.3	33	2	16.6
AVERAGE			25.3 (mins)			42.1 (mins)

Table 7: ON-Line Spent Effort by Call-Type

call-type	SYS_1			SYS_2		
	time (mins)	problems	time/problem	time (mins)	problems	time/problem
clarify document	685	35	19.6	305	7	43.6
operation question	2317	46	50.4	4062	78	52.1
pre-sales request	45	2	22.5	130	2	65.0
capability, feature	1105	30	36.8	855	17	50.3
other	240	13	18.5	1810	19	95.3
document fault	117	1	117.0	-	-	-
operation fault	210	10	21.0	75	20	3.8
application SW fault	330	1	330.0	-	-	-
system SW fault	1125	15	75.0	769	4	192.3
system SW change request	115	3	38.3	335	2	167.5
instruction fault	20	2	10.0	-	-	-
HW fault	780	17	45.9	65	2	32.5
AVERAGE			40.5 (mins)			55.7 (mins)

Table 8: OFF-Line Spent Effort by Call-Type

- by the number of phone calls per problem:

  The average number of calls (interactions with the customer) per problem is about 4 (SYS_1: 3.7, SYS_2: 4.1) according to table 4.

  The most crucial problems in SYS_1 in terms of number of calls are: documentation faults (7 calls per problem), operation faults (5 calls per problem), and system software faults (5.7 calls per problem). In the case of SYS_2, the most crucial problems are: documentation clarifications (4.9 calls per problem), operation requests (4.8 calls per problem), pre-sales requests (4.5 calls per problem), and capability/feature requests (4.9 calls per problem).

- by the effort spent on-line (time spent talking to the customer on the phone ---> Table 7):

  The average effort per problem spent on-line is about 30 minutes.

  In the case of SYS_1, most on-line effort is spent for documentation problems (43 minutes per problem), application software faults (53 minutes per problem), and system software faults (56 minutes per problem). In the case of SYS_2 most on-line effort is spent for pre-sales requests (105 minutes per problem)

- by the effort spent off-line (time spent other than talking to the customer on the phone ---> Table 8):

  The average effort per problem spent off-line is about 45 minutes.

  In the case of SYS_1, the most off-line effort is spent for documentation problems (117 minutes per problem) and application software faults (330 minutes). In the case of SYS_2, the most off-line effort is spent for system software faults (180 minutes per problem).

(G2) Identify the best ways of applying the 20/80 rule to get the biggest savings and return on our maintenance dollars:

Although we have no final results concerning this matter, a careful interpretation of the results related to goal (G1) indicates that for instance better documentation, in the case of SYS_1, could save a big percentage of maintenance problems. In a paper not related to this study an analysis of software maintenance changes is reported[10]; the authors aim at the development of metrics for predicting where those changes might occur. Such metrics might help save dollars by concentrating resources on subsystems or modules which can be expected to require many changes.

(G3) Identify criteria for when a product is ready for release:

This question can only be answered if we know more about the type of problems and effort spent in engineering before release (question Q4) and about the type and problems during field test (question Q5).

(G4) Identify features of product, documentation or support that provide a wider customer satisfaction:

This question can be addressed by designing a customer questionnaire. Some of the technical problems definitely have impact on the customer's satisfaction, such as the high number of documentation-related problems (in the case of SYS_1) or not being able to keep promised dates for calling customers back.

(G5) Identify criteria for when a software product should be rewritten rather than maintained:

Unfortunately there are no data collected indicating explicitly which parts (modules, subsystems) of a product were affected (question Q26) or whether a problem is due to a previous change (question Q8).

The only way to address this question by using the currently available data is to evaluate the actual patch where the actual lines changed are listed. A paper not related to this study indicates that complexity metrics characterizing the locality of changes might be a promising metric for characterizing the suitability of parts of a software system for maintenance purposes[11].

(G6) Identify metrics of customer satisfaction that can be developed based upon existing data:

Based upon the results concerning goal G4 we hope to be able to develop metrics for customer satisfaction. Although it is too early to expect reliable metrics, candidate metrics might include aspects such as ability to keep promised schedules for dealing with maintenance problems or the frequency of similar (at least from the customer's point of view) maintenance problem reports.

(G7) Develop organizational guidelines for integrating software quality metrics into the company's framework of design, development, and support:

This goal represents the second step after having understood the maintenance problems and identified possible improvements. Procedures for monitoring quality and productivity have to be established throughout the development and maintenance of software products; the prescribed data and metrics should be used for management and motivation purposes and improved. Before this problem can be addressed in a satisfactory way many more and different analyses have to be performed; in particular, data concerning the development phase of products have to be collected in order to identify the impact of the particular development process on maintainability. In a paper not related to this study interesting approaches for predicting the required customer support for a particular system were presented[8]. The prediction approach utilized development metrics among others.

9. Measurement and Evaluation Tools

In order to apply the proposed quantitative assessment approach practically, data collection and validation procedures as well as evaluation procedures need to be automated. A tool system was proposed integrating many tools already available in this environment. The whole tool system needs to be implemented in a decentralized fashion around a central data base. It has to provide different interfaces to different maintenance groups, limiting each group only to data relevant to their specific task, presenting the data in a helpful way. Independent of this company-specific project, a research project at the University of Maryland is aiming at the development of a comprehensive approach to automating measurement and evaluation in the context of software projects which include support of the generation of goals and questions and the project-specific interpretation of measurement results[2,4].

10. Conclusions

The objective of this study has been to demonstrate the benefits of assessing the software maintenance process in a quantitative way for the purpose of improvement. We have been able to show the applicability of the goal/question/metric paradigm to this complex problem domain and derive first analysis results based on a very limited subset of available data. The long-range benefits can be expected to be much more significant provided the derived set of data are collected in the future and interpreted within the proper context of maintenance questions and goals. In this paper we have not addressed the psychological problems involved in trying to introduce quantitative approaches into a traditional maintenance environment. The interested reader is referred to a book describing Hewlett Packard's experience (including psychological problems of motivating project personnel and higher-level management) from introducing metrics into their daily software production environment[9].

It was even surprising to us, how many characteristics of the maintenance process could be made visible by analyzing the limited set of data available at the time. This visibility of characteristics might be helpful in communicating problems in a more objective and convincing way.

The analysis result underline the importance of viewing software maintenance not as an isolated activity but as integrated into the overall software life cycle. We can improve the effectiveness of maintenance procedures by purely analyzing the maintenance process. However, we will never reduce the overall effort (and money) spent for maintenance below a certain limit if we cannot make sure that software products fulfill certain quality requirements at the time of delivery (start of maintenance). Low quality products will always cause maintenance problems. Accepting this fact will lead us to establish quality criteria for a product to be released to customers and, thereby, entering the maintenance phase. As a consequence, developers could develop guidelines for how to achieve those criteria and metrics to evaluate the degree to which those criteria are actually met. Altogether this would allow us to develop better maintainable products in the first place or, at least, allow us to predict certain maintenance problems at the beginning of maintenance. Additional benefits of collecting maintenance data are to provide a better basis for judging customer satisfaction, the company's image, and marketing.

If we want to reduce the overall maintenance effort we need to apply the assessment and improvement procedures introduced in this paper to development as well as maintenance of a product. This requires the availability of development data (as implicated by the evaluation questions in section 4) in addition to maintenance data. As long as we do not assess the overall software life cycle, problems will shift from design to coding, coding to testing, and development to maintenance. It is a well known fact that the really serious maintenance problems originate during the prior development of the product; the identification of these real causes of maintenance problems will result in significant improvements of maintenance.

Acknowledgements

This study was supported by a grant from Burroughs Corporation to the University of Maryland. The following people from Burroughs and System Development Corporation supported this study in various ways: Kenneth Cain, Charles H. Coulbourn, Joe Dormady, Al Freund, Michael A. Heneghan, Bob Lesniowski, Mary Mikhail, Frank Star, and Christopher Whitener. Jinn-Ke Jan from the University of Maryland worked on this project as a graduate research assistant.

References

[1] V. R. Basili, "Quantitative Evaluation of Software Engineering Methodology," First Pan Pacific Computer Conference, Melbourne, Australia, September 1985 [also available as Technical Report, TR-1519, Dept. of Computer Science, University of Maryland, College Park, July 1985].

[2] V. R. Basili, H. D. Rombach, "TAME: Tailoring an Ada Measurement Environment," Proceedings of the Joint Ada Conference, Arlington, Virginia, March 16-19, 1987, pp. 318-325.

[3] V. R. Basili, H. D. Rombach, "Tailoring the Software Process to Project Goals and Environments," Proceedings of the Ninth International Conference on Software Engineering, Monterey, California, March 30 -April 2, 1987, pp. 345-357.

[4] V. R. Basili, H. D. Rombach, "TAME: Integrating Measurement into Software Environments," submitted for publication to IEEE Transactions on Software Engineering.

[5] Victor R. Basili, Richard W. Selby, Jr., "Data Collection and Analysis in Software Research and Management," Proc. of the American Statistical Association and Biometric Society Joint Statistical Meetings, Philadelphia, PA, August 13-16, 1984.

[6] V. R. Basili, D. M. Weiss, "Evaluating Software Development by Analysis of Changes: The Data from the Software Engineering Laboratory," Technical Report TR-1236, Dept. of Computer Science, University of Maryland, College Park, December 1982.

[7] V. R. Basili, 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.

[8] J. Chapin, G. Faidell, "Predicting Software Customer Support," Proc. Conference on Software Maintenance, Washington D.C., 1985, pp. 128-134.

[9] Robert B. Grady, Deborah L. Caswell, "Software Measures; Establishing a Company-Wide Program," to be published as a book by Addison Wesley, 1987.

[10] D. A. Gustafson, A. Melton, and C. S Hsieh, "An Analysis of Software Changes During Maintenance and Enhancement," Proc. Conference on Software Maintenance, Washington D.C., 1985, pp. 92-95.

[11] H. Dieter Rombach, "A Controlled Experiment on the Impact of Software Structure on Maintainability," IEEE Transactions on Software Engineering, Vol. SE-13, No. 3, March 1987, pp.344-354.