

USAISEC

US Army Information Systems Engineering Command
Fort Huachuca, AZ 85613-5300

AD-A268 100



2

U.S. ARMY INSTITUTE FOR RESEARCH
IN MANAGEMENT INFORMATION,
COMMUNICATIONS, AND COMPUTER SCIENCES

DTIC
ELECTE
AUG 16 1993
S **D**
C

The Software Support Qualitative Assessment Methodology

Volume IV

Implementing the Support Organization Assessment Measure

DISTRIBUTION STATEMENT A

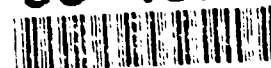
Approved for public release
Distribution Unlimited

Best Available Copy

March 1991

ASQB-GI-91-019

93-18746



AIRMICS
115 O'Keefe Building
Georgia Institute of Technology
Atlanta, GA 30332-0800



UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0186
Exp. Date Jun 30, 1996

1a REPORT SECURITY CLASSIFICATION UNCLASSIFIED			1b RESTRICTIVE MARKINGS NONE	
2a SECURITY CLASSIFICATION AUTHORITY N/A			3 DISTRIBUTION / AVAILABILITY OF REPORT N/A	
2b DECLASSIFICATION / DOWNGRADING SCHEDULE N/A				
4 PERFORMING ORGANIZATION REPORT NUMBER(S) ASQB-GI-91-019			5 MONITORING ORGANIZATION REPORT NUMBER(S) N/A	
6a NAME OF PERFORMING ORGANIZATION The Center for Information Management Research		6b OFFICE SYMBOL (if applicable)	7a NAME OF MONITORING ORGANIZATION N/A	
6c ADDRESS (City, State, and ZIP Code) SERC Georgia Institute of Technology Atlanta, GA 30332		Dept of MIS University of Arizona Tucson, AZ 85613	7b ADDRESS (City, State, and Zip Code) N/A	
8a NAME OF FUNDING/SPONSORING ORGANIZATION AIRMICS		8b OFFICE SYMBOL (if applicable) ASQB - GI	9 PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER	
8c ADDRESS (City, State, and ZIP Code) 115 O'Keefe Bldg., Georgia Institute of Technology Atlanta, GA 30332-0800		10. SOURCE OF FUNDING NUMBERS		
		PROGRAM ELEMENT NO	PROJECT NO	TASK NO
		62783A	DY10	02-01-01
11 TITLE (Include Security Classification) The Software Support Qualitative Assessment Methodology Volume IV (UNCLASSIFIED) Implementing the Support Organization Assessment Measure				
12 PERSONAL AUTHOR(S) W. Michael McCracken, Elizabeth Mynatt, Christopher Smith (GIT) J.F. Nunamaker, Ai-Mei Chang, Titus Purdin, Richard Orwig, Amit Vyas (Univ of Arizona)				
13a TYPE OF REPORT final report		13b. TIME COVERED FROM _____ TO _____		14 DATE OF REPORT (Year, Month, Day) 1991, March, 22
15 PAGE COUNT 48				
16 SUPPLEMENTARY NOTATION				
17 COSATI CODES			18 SUBJECT TERMS (Continue on reverse if necessary and identify by block number)	
FIELD	GROUP	SUB-GROUP	life cycle metrics, systems support; support measures; qualitative assessment; supportability measures, assessment measures; readiness measures, information systems; software maintenance; support tools; tools management;	
19 ABSTRACT (Continue on reverse if necessary and identify by block number)				
The Software Supportability Qualitative Assessment Methodology is a five volume reference set that provides measures to aid in the support of information systems. The volumes are aimed at improving the support process by more accurately assessing the capabilities of support organizations, qualitatively measuring the supportability of fielded systems and evaluating the operational readiness of fielded systems. The five volumes are: I. Developing Quality Measures for Information Systems Support II. The Review of Metrics for Developing an Information Systems Support Measurement Framework III. Implementing the Software Supportability Measure IV. Implementing the Support Organization Assessment Measure V. Implementing the Operational Readiness Measure This volume provides instructions for collecting data for the assessment, conducting the assessment, and interpreting the final results. Also included are guidelines for improving the capabilities of a support organization based on its evaluation.				
20 DISTRIBUTION / AVAILABILITY OF ABSTRACT ☒ UNCLASSIFIED / UNLIMITED ☐ SAME AS RPT ☐ DTIC USERS			21 ABSTRACT SECURITY CLASSIFICATION UNCLASSIFIED	
22a NAME OF RESPONSIBLE INDIVIDUAL Howard C "Butch" Higley		22b TELEPHONE (Include Area Code) (404) 894-3110		22c OFFICE SYMBOL ASQB-GI

The research herein was performed for the Army Institute for Research in Management Information, Communications, and Computer Sciences (AIRMICS), the RDTE organization of the U.S. Army Information Systems Engineering Command (USAISEC). The sponsor for the project was the Office of the Director of Information Systems for Command, Control, Communications, and Computers (ODISC4). The principal investigator was from The Center for Information Management Research (CIMR), W. Michael McCracken of the Georgia Institute of Technology.

This research report is not to be construed as an official Army position, unless so designated by other authorized documents. Material included herein is approved for public release, distribution unlimited, and is not protected by copyright laws. Your comments on all aspects of the document are solicited.

Accession For	
NTIS CRA&I	☒
DTIC TAB	☐
Unannounced	☐
Justification	
By <i>Rec Fosen 50</i>	
Distribution /	
Availability Codes	
Dist	Avail and/or Special
<i>A-1</i>	

DTIC QUALITY INSPECTED 3

THIS REPORT HAS BEEN REVIEWED AND IS APPROVED

s/ _____
Glenn E. Racine
Chief
CISD

s/ _____
John R. Mitchell
Director
AIRMICS

**The Software Support
Qualitative Assessment Methodology
Volume IV
Implementing the
Support Organization Assessment Measure**

Prepared by
The Center for Information Management Research
for the
U.S. Army Institute for Research in Management
Information, Communications, and Computer Science
(AIRMICS)
Contract No. ECD-8904815

*J. F. Nunamaker Jr., Ai-Mei Chang, Titus Purdin
Richard Orwig, Amit Vyas*
Department of Management Information Systems
University of Arizona

December 1990

The **Software Supportability Qualitative Assessment Methodology** is a five volume reference set that provides measures to aid in the support of information systems. These manuals are aimed at improving the support process by more accurately assessing the capabilities of support organizations, quantitatively measuring the supportability of fielded systems and evaluating the operational readiness of fielded systems.

Volume I, *Developing Quality Measures for Information Systems Support*, describes the three measures along with the model of information system support that the measures are designed to satisfy. This is the main volume of the set and should be consulted before implementing the measures described in more detail in the other volumes.

Volume II, *The Review of Metrics for Developing an Information Systems Support Measurement Framework*, provides a survey and evaluation of current metrics in terms of information systems support. Specifically, three classes of metrics are reviewed: software product metrics, life cycle process metrics, and process management metrics.

Volume III, *Implementing the Software Supportability Measure*, provides instructions for collecting data for the measure, compiling the measure by evaluating the data, and interpreting the final result. The volume also contains guidelines for improving the supportability of an information system based on its evaluation. Specifically, the volume contains resource estimations for compiling and evaluating the measure, questionnaires for collecting the required data and step-by-step instructions for measuring the supportability of an information system.

Volume IV, *Implementing the Support Organization Assessment Measure*, provides instructions for collecting data for the assessment, conducting the assessment, and interpreting the final result. The volume also contains guidelines for improving the capabilities of a support organization based on its evaluation. Specifically, the volume contains resource estimations for conducting and evaluating the assessment, questionnaires for collecting the required data and step-by-step instructions for measuring the capabilities of a support organization.

Volume V, *Implementing the Operational Readiness Measure*, provides instructions for collecting data for the measure, compiling the measure by evaluating the data, and interpreting the final result. The volume also contains guidelines for improving the operational readiness of an information system based on its evaluation. Specifically, the volume contains resource estimations for compiling and evaluating the measure, questionnaires for collecting the required data and step-by-step instructions for measuring the operational readiness of an information system.

Contents

1 Introduction	2
2 Questionnaire Usage Guide	3
Who Should Fill It Out	3
Material	3
Scoring Procedure	4
3 Suggestions for Process Improvements	5
A Glossary of Terms	7
B List of Acronyms	10
C Organization Assessment Questionnaire	11
D Matrix	12
E Assessment Recording Form	13
F User's Guide	14

1 Introduction

The support organization assessment measure is focused on defining the important positive factors that characterize an effective support organization. The objective of the measure is to position an organization in one of the five levels of support process maturity based on that organization's recognition and practice of these factors. The measure can also highlight problem areas where improvements are needed.

This method is patterned after the assessment method developed by the Software Engineering Institute (SEI) at Carnegie Mellon University. SEI measures the quality of software systems development by using a questionnaire to elicit information about an organization's knowledge and practice of software development factors. Each question on this questionnaire is weighted according to its position along a five-point maturity scale. The questionnaire is filled out by members of the development organization in conjunction with representatives of SEI. The answers are evaluated and the organization is assigned a maturity score between one and five that correspond to five levels of software development maturity: initial, repeatable, defined, managed, and optimized.

The SEI Assessment is a measure of an organization's maturity with respect to the software development process. This organizational assessment with respect to the support process is different from SEI's assessment of software development because the software support process is different than the software development process. The factors that determine a mature software development organization are not necessarily the same factors that determine a mature software support organization. Many factors are shared by the two processes but the weights of the common factors will be different depending upon whether a maturity measurement is desired or whether a measurement of software development is desired.

Thus, the assessment measure involves having a representative or representatives of the support organization answer questions that indicate that organization's awareness and practice of the software support factors with respect to the *software support* process. Each question is categorized as a Level 1, 2, 3, 4, or 5 question. Each question requires a "yes" or "no" response. The answers are tallied and an evaluation is made by determining the percentage of "yes" answers that are made with respect to the questions in each level. A Level 1 organization will answer "yes" to 80% of the Level 1 questions. A Level 2 organization will answer "yes" to 80% of all Level 1 and Level 2 questions; a Level 3 organization will answer "yes" to 80% of all Level 1, 2 and 3 questions; and so on.

The questionnaire that we have developed for use for the organizational assessment is in Appendix C. The steps involved in developing this questionnaire are:

1. Collect support organization assessment factors from the organizational perspective.
2. Categorize these factors.
3. Devise "yes" or "no" questions that elicit respondent's knowledge of or adherence to these factors.
4. Weigh the questions with respect to software support maturity levels.
5. Place these questions on a Software Support Maturity Matrix.

Appendix D shows the matrix with maturity levels across the chart and organizational support factors on the vertical axis. Each cell of the matrix lists those question numbers that elicit information pertaining to an organization's awareness and practice of a factor of software support at a certain level of maturity. For instance, question 1.1.8, "Does the Software Maintenance Quality Assurance function have a management reporting channel separate from the software project management?", is the eighth question dealing with the factor of organizational structure. It is a level 3 question in that, by answering "yes" to this question, the organization has used the organizational structure as a method to ensure that those responsible for the quality of software support are not controlled by those people that are tasked with project completion. The chart (Appendix D) shows the distribution of questions that are used to place an organization at a particular level.

As previously noted, the overall maturity level of an organization is based upon the percentage of affirmative answers to each question at each level of maturity for all categories of questions. However, an organization may be a Level 3 organization with respect to organizational issues, at Level 2 for software support process and personnel factors, and at Level 1 with respect to factors pertaining to tools and technology. The overall assessment may result in a Level 2 score. This suggests that the organization would want to become more aware of tools and technologies that may be applicable to the support process of that organization in order to provide better software support.

2 Questionnaire Usage Guide

The primary goal of the questionnaire is to help a support organization evaluate its support capabilities. In combination with this evaluation, the responses to the questions will suggest factors that will help the organization improve the quality of the support that the organization provides.

Who Should Fill It Out

The intention of the questionnaire is that it is a self-assessment rather than an assessment performed by an outside agency. The specific person most appropriate for filling out this questionnaire is the manager or director of the support organization tasked with supporting a portfolio of software applications. The idea of the questionnaire is not to point fingers at any one organization, but to know for oneself as to how one's own performance is. This the reason why this questionnaire needs to be filled by the manager rather than anyone else. The manager can get help from users and software developers and some selected staff to aid in answering these questions truthfully. If used as specified, the questionnaire can uncover problem areas in addition to identifying the level at which the organization is.

Material

Four items comprise the material required to perform the assessment: the questionnaire, the answer sheet, the evaluation guide and the glossary.

The questionnaire contains a series of questions that are answered "yes" or "no." The

questions are categorized according to issues pertaining to management of the organization, software process, technology, and personnel. Each question concerns an important factor of the support organization assessment measure that is related to category in which it is contained. Many of the questions may seem similar. However, in order to properly answer a question the purpose of that question must be viewed in the perspective of the category in which it is located.

The answer sheet is a form used to mark the answers to the questions. It is also used as a worksheet to determine the level at which the organization is operating (ad-hoc, repeatable, methods, control, or optimal).

Because of differences in terminology and differing software support organizations it is necessary to include a glossary of terms that help in understanding the intention of some of the questions on the questionnaire. Because of the nature of the "yes" and "no" questions, it is most important that the person filling out the questionnaire understands the terminology used in the questions and the intention of each individual question. As mentioned earlier, questions may seem very similar. Persons filling out this questionnaire should first ascertain that they are viewing questions with respect to the category in which they are contained. They should also freely consult the glossary for any terms that they feel are ambiguous or unclear.

The evaluation guide is a short step by step procedure for determining the overall measure of the quality of software support supplied by the organization. This guide is used with the answer sheet to determine that level at which an organization supports the software applications in its portfolio.

Scoring Procedure

In order to rank support organizations, five maturity levels have been determined. These levels represent evolving stages for software support organizations. An organization at the most rudimentary stage of software support would be at level 1, the most sophisticated as predicted by researchers would be at level 5, and the remaining maturity levels would indicate different degrees of evolutionary growth and capability. Ranking an organization consists of determining an overall maturity level based upon the organization's knowledge of and adherence to the factors associated with the software support process. For example, a level 3 organization would know and adhere to almost all of the factors associated with levels 1, 2 and 3.

The questions are designed to allow for easy scoring of individual questions. In order to determine the area-wise ranking and corresponding level, a procedure requiring successive qualification at each level is used. This is as follows:

1. Determine the percentage of affirmative answers to all Level 1 questions, and if this is at least 80% the organization has qualified for assessment for level 2, or else it is at level 1.
2. Determine the percentage of affirmative answers to all Level 2 questions, and if this is at least 80%, the organization has qualified for assessment for level 3, or else it is at level 2.

3. Determine the percentage of affirmative answers to all Level 3 questions, and if this is at least 80%, the organization has qualified for assessment for level 4, or else it is at level 3.
4. Determine the percentage of affirmative answers to all Level 4 questions, and if this is at least 80%, the organization is at level 5, or else it is at level 4.

A classification for overall maturity may be obtained by following the same procedure outlined above with the entire set of questions. This overall classification helps to provide a complete picture of the support capability. Overall quality software support requires a balance of maturity in all four categories of support organization assessment factors (see Figure 1). However area-wise classification helps highlight specific strengths and weaknesses of the organization. The implications of the different levels are as per the descriptions in the previous section discussing Levels of Software Supportability.

3 Suggestions for Process Improvements

The levels of assessment of the support organization provide a spectrum of maturity levels. As such, a Level 1 organization cannot become a Level 5 organization without first becoming a Level 2, 3, and 4 organization.

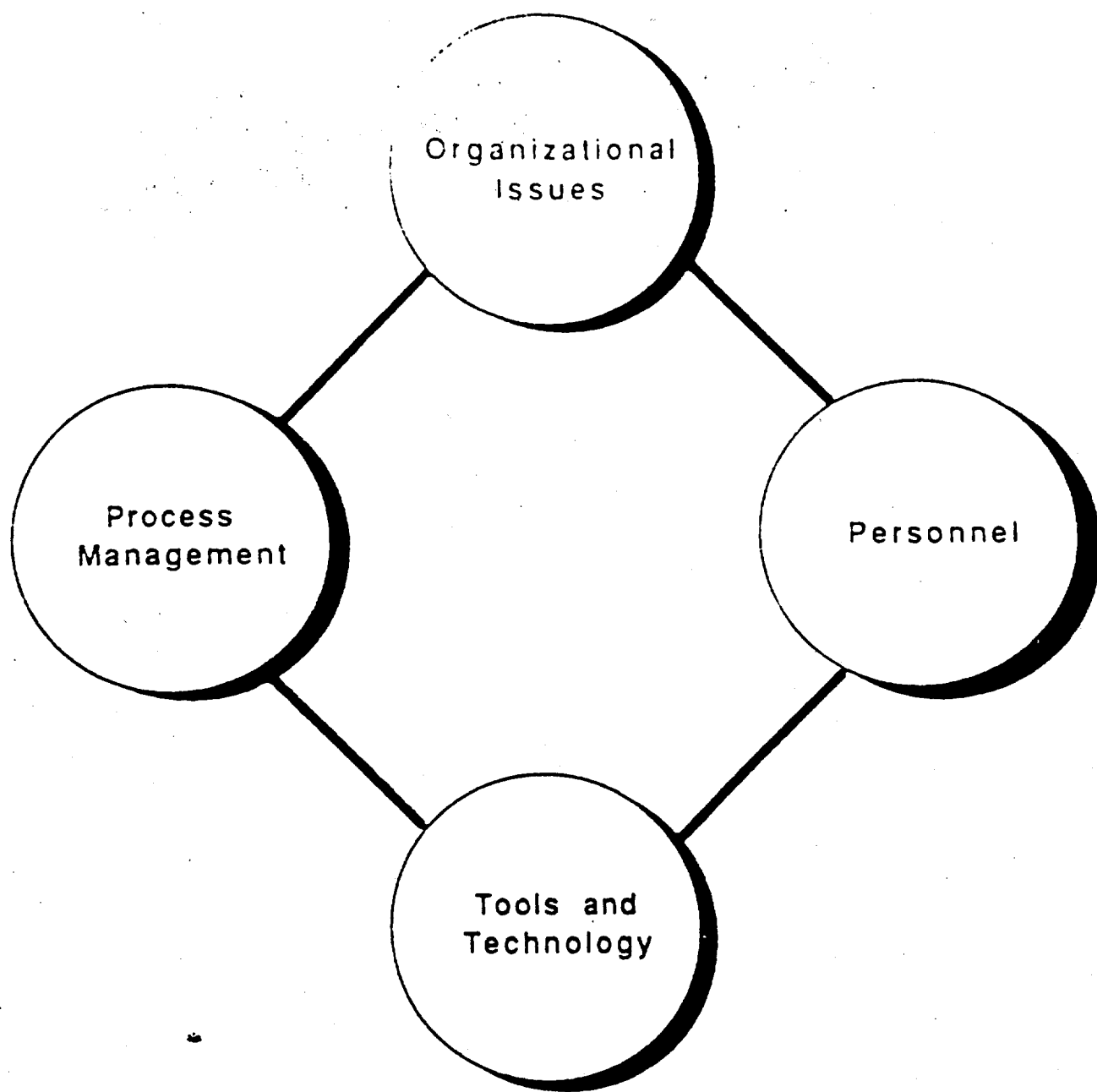
To grow from an ad-hoc, fire-fighting level of Level 1 to Level 2 an organization must have procedures which allow for the collection, evaluation, categorization, and prioritization of software problems and planning mechanisms for determining the schedule of the fixes.

Level 2 organizations can be counted on to provide accurate estimates of problem fixes for those systems that they support but cannot be depended upon for any new systems that they may have assigned to them. In order for a Level 2 organization to achieve Level 3 capabilities, it must have enough knowledge of the software support process itself that it has defined the methods of software support. This usually involves have full-time people devoted to supporting the software support process instead of only providing software support. The steps, methods and procedures used by the organization need to be documented.

To reach Level 4, a Level 3 organization must actually provide measurements to indicate that the steps, methods and procedures that it has documented at Level 3 are actually followed. Minimum measures are set and actual costs and benefits can be quantified.

At Level 5, the costs and benefits are quantified, recorded, and compared with past performance to determine which policies, procedures, and resources are best used in particular circumstances. The optimal software support organization completely understands the support process and has efficient policies and procedures in place to effectively manage the resources available to perform the corrective, adaptive, and perfective maintenance requirements of the portfolio of application systems that it has to support.

One organization may perform at different maturity levels among the separate categories of factors. For example, a hypothetical organization may be a Level 3 organization with respect to organizational issues, at Level 2 for software support process and personnel factors, and at Level 1 with respect to factors pertaining to tools and technology (see Figure 2). The overall assessment may result in a Level 2 score. This suggests that the

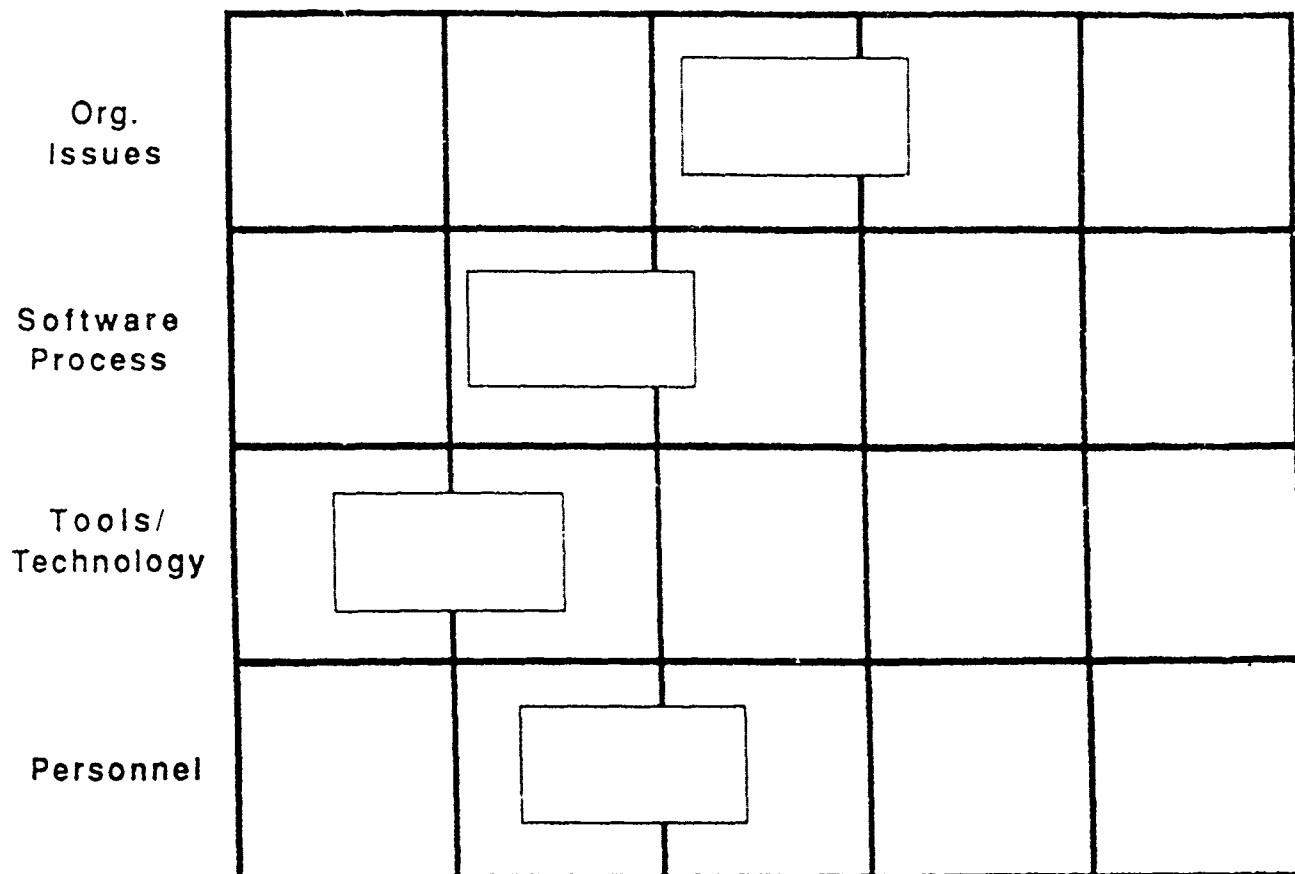


SUPPORT ORGANIZATION ASSESSMENT FACTORS

FIGURE 1

LEVELS

1	2	3	4	5
Ad-hoc	Repeatable	Methods	Control	Optimal



Hypothetical Organization Ranking

FIGURE 2

organization needs to become more aware of tools and technologies that may be applicable to the support process of that organization in order to provide better software support.

The most immediate improvement for this hypothetical organization will come from analyzing the "no" responses of those questions that fall into the Level 2 categories of software support process and personnel and the Level 1 and 2 tools and technology category. This provides specific factors that will help the organization mature in a smooth fashion. It would not make sense for this organization to concentrate on Level 5 questions for any category without having considered questions contained in Levels 3 and 4 first.

A Glossary of Terms

Acceptance Review A review of a software product by developers and maintainers to determine if the product satisfies all originally specified requirements.

Acceptance Test Testing led by the client or QA group to determine whether the product satisfies its specifications as claimed by the developer.[Sch90]

Application System same as Information System

Availability A measure of the degree to which an item is in an operable and committable state at the start of a mission when the mission is called for at a random point in time.[Dep82]

Benchmark Testing Evaluation of the system performance against quantitative requirements.[Sch90]

Change Request Review Board An authority responsible for evaluating and approving requests for changes to a software product.

Cohesion A measure of the degree of the functional relatedness within program units. [Som89]

Complexity A characteristic of the software interface which influences the resources another system will expend or commit while interfacing with the software. [CDS86]

Configuration Management The process of identifying and defining the configuration items (hardware/software units) in a system, controlling the release and change of these items throughout the system life cycle, recording and reporting the status of configuration items and change requests, and verifying the completeness and correctness of configuration items.[IEE83]

Consistency The extent to which uniform design techniques and notation are used. [War87]

Coupling A measure of the strength of interconnections (dependencies) between program units. [Som89]

Error Human action that results in software containing a fault. Examples include omission or misinterpretation of user requirements in a software specification, incorrect translation or omission of a requirement in the design specification. [IEE83]

Failure A departure of program operation from program requirements.[IEE83]

Failure Rate The number of failures of an item per measure-of-life unit.[Dep82]

Fault A manifestation of an error in software. A fault, if encountered, may cause a failure. Synonymous with bug.

Fourth Generation Language (4GL) A computer programming language that provides abstractions of data and/or procedural specifications and is usually suited for a particular application domain.

Integration Testing Verify that the modules of the system combine correctly in order to achieve a product that meets its specifications. [Sch90]

IS (Information Systems) Organization An organized collection of procedures, personnel, and resources dedicated to support a portfolio of information systems.

Lines of Code Lines of source code, not including comments.

Maintainability The probability that an item will be retained in, or restored to, a specified condition within a given period if prescribed procedures and resources are used.[Dep82]

Maintenance All actions required to retain an item in, or restore it to, a specified condition.[Dep82]

Maintenance Audit An organized review of the maintenance organization.

Maintenance Escort Participation of the software maintainer in software system development.

Man/Machine Interface The software that supports the interaction between the user and the system.

Measure A high-level unit of specification which characterizes, evaluates, or predicts various aspects of software life cycle processes and products.

Metric A measurable indication of some aspect of a system. [DeM82] A quantification of a specific feature of the software life cycle process or software product.

Modularity A characteristic of software such that it is well-structured, highly cohesive, and minimally coupled. [War87]

New Systems Development The development of a system which has never been fielded.

Object Oriented Design Designing a system in terms of abstract data types where the objects are instantiations of the data types and new data types can be defined as extensions of previously defined types.

Regression Testing Testing the system against previous test cases to ensure that the functionality of the system has not been compromised by recent changes to the system. [Sch90]

Reliability The probability that an item will perform its intended function for a specified interval under stated conditions.[Dep82]

Self-Descriptiveness A characteristic of software that enables the understanding of implementation of software functions. [War87]

Support Staff The personnel tasked with maintaining an information system.

Supportability A measure of the adequacy of products, resources, and procedures to facilitate the support activities of modifying and installing software, establishing an operational software baseline, and meeting user requirements. [PTH87]

Testability The extent to which software facilitates both the establishment of test criteria and the evaluation of the software with respect to those criteria. [IEE83]

Throw-away prototyping Creating a prototype as part of system design and then "throwing away" the prototype and implementing the system "from scratch" not using any of the source code from the prototype.

Top-down design Designing the system by recursively breaking the system down into smaller components.

Unit Testing Testing of individual portions of the system.

B List of Acronyms

AIRMICS U.S. Army Institute for Research in Management Information, Communications, and Computer Science

AMC Army Materiel Command

CCB Change Control Board

COE Army Corps of Engineers

FORSCOM Forces Command

HSC Army Health Services Command

IS Information System

ISC Army Information Systems Command

LOC Lines of Code

C Organization Assessment Questionnaire

This appendix contains a 12 page questionnaire (not including the questionnaire cover page) for gathering organizational assessment data. The questionnaire should be photocopied and distributed to selected respondents.

SOFTWARE SUPPORTABILITY QUALITATIVE
ASSESSMENT METHODOLOGY

ORGANIZATION ASSESSMENT
QUESTIONNAIRE

Software Supportability Qualitative Assessment Methodology
Organization Assessment Questionnaire

ORGANIZATIONAL ISSUES

1.1 Organizational Structure:

	<u>YES</u>	<u>NO</u>
1.1.1. Are departments/groups in the software organization structured on the basis of <u>life cycle phase</u> (separate Development and Maintenance groups)?	—	—
1.1.2. Are departments/groups in the software support organization structured on the basis of <u>application skill</u> (financial applications payroll applications etc.)?	—	—
1.1.3. Are departments/groups in the software organization structured on the basis of <u>technical skills</u> (separate groups for systems analysts, programmers)?	—	—
1.1.4. Do the same software personnel perform both development and maintenance functions?	—	—
1.1.5. Are there specific measures currently used for determining the effectiveness of your software maintenance organization?	—	—
1.1.6. For each project involving software maintenance, is there a designated software manager?	—	—
1.1.7. If so, does this software manager report to an overall project manager?	—	—
1.1.8. Does the Software Maintenance Quality Assurance function have a management reporting channel separate from the software project management?	—	—
1.1.9. Is there a designated individual or team responsible for the control of software interfaces (refer to question in tools section 3.2.10)?	—	—
1.1.10. Is software system engineering represented on the software maintenance team?	—	—
1.1.11. Is there a software configuration control function for each software maintenance project?	—	—

Software Supportability Qualitative Assessment Methodology
Organization Assessment Questionnaire

	<u>YES</u>	<u>NO</u>
1.1.12. Is there a formal procedure or plan for co-ordinating tasks within the support staff?	—	—
1.1.13. Is there a specific group that determines software maintenance procedures?	—	—
1.1.14. Are there formal job descriptions available for each member of the staff?	—	—
 1.2 Portfolio Characteristics and Management:		
1.2.1. Are there fewer than five different source languages that you support in the application portfolio?	—	—
1.2.2. Are profiles maintained of sizes of applications in the application portfolio?	—	—
1.2.3. Are profiles maintained of the resources expended per application in the application portfolio?	—	—
 1.3 Physical Facilities:		
1.3.1. Does each maintenance programmer have adequate access to appropriate computing facilities?	—	—
1.3.2. Can you emulate the user hardware and software configurations for each application in your portfolio?	—	—
 1.4 Budgetary Control:		
1.4.1. Is at least 30% of the maintenance budget spent on improving maintenance Quality?	—	—
1.4.2. Is there a separate budget for maintenance and development projects?	—	—
1.4.3. Is there a separate budget for each application that is supported in the support portfolio?	—	—

Software Supportability Qualitative Assessment Methodology
Organization Assessment Questionnaire

YES NO

1.5 Effectiveness of the organization as a whole:

1.5.1. Is the organizational effectiveness predominantly determined by the size of the change request backlog?

1.5.2. Is there a mechanism for user evaluation of the support organization's effectiveness?

1.5.3. Would you perceive the quality and the effectiveness of the support function that you provide to be:

(i) poor?

(i) fair?

(iii) good?

(iv) excellent?

Relationship with User Organization:

1.6.1. Do you consider 90% of the user population of your application portfolio to be 'computer literate'?

1.6.2. Do you keep a record of all communications between your organization and user organizations?

1.6.3. Are over 50% of the communications between your support organization and the users initiated by your organization?

1.6.4. Do you have a newsletter or other regular vehicle for communicating with your user organizations?

1.6.5. Is there a formal mechanism for discussing/negotiating change requests initiated and their impacts with the users?

1.6.6. Do the users consider the quality and effectiveness of the support function as good?

1.6.7. Would you say that the current application staff communicates directly with the users of the applications:

(i) daily?

(ii) not daily but atleast weekly?

(iii) not weekly but atleast monthly?

Software Supportability Qualitative Assessment Methodology
Organization Assessment Questionnaire

YES NO

1.6.8. Would you concur that the users reasonably understand the application system they use (or are they experienced in the applications they use)?

— —

1.6.9. Do you think the user expectations (in terms of change requests and enhancements) are realistic?

— —

1.7 Relationship with the Development Organization:

1.7.1. Is there a formal channel for communication between the maintenance personnel and the developers of the software with regard to maintainability of the software:

(i) during the development stage of the software?

— —

(ii) after the installation of the software?

— —

SOFTWARE SUPPORT PROCESS AND ITS MANAGEMENT

2.1 Standards and Procedures:

2.1.1. On each supported software system, does your organization have a set of standardized and documented procedures to follow in:

(i) modifying system code?

— —

(ii) testing system code?

— —

(iii) using specific tools and techniques for using them?

— —

2.1.2. Are *formal procedures* used for

(i) estimating sizes/extent of changes to systems?

— —

(ii) estimating software maintenance cost (over a given period of time)?

— —

(iii) tracking the size of software system(s) being maintained?

— —

2.1.3. Is a *mechanism* used for ensuring that the software maintenance team becomes familiar with the system being maintained?

— —

2.1.4. Are standards applied to software in a maintenance project?

— —

2.1.5. Are standards applied to the preparation of unit test cases?

— —

2.1.6. Are re-design review standards applied?

— —

Software Supportability Qualitative Assessment Methodology
Organization Assessment Questionnaire

	<u>YES</u>	<u>NO</u>
2.1.7 Are there <i>formal</i> organizational <i>procedures</i> for prioritizing the incoming change requests?	—	—
2.1.8. Are there <i>formal</i> organizational <i>procedures</i> for prioritizing the maintenance work with competing demands for time?	—	—
 2.2 Process Metrics:		
2.2.1. Are records of manpower expenditures maintained for actual vs. planned software support?	—	—
2.2.2 Are <u>records</u> of:		
(i) planned vs. actual maintenance completion dates maintained?	—	—
(ii) planned vs. actual testing completion durations maintained?	—	—
(iii) software units maintained (over a given period of time)?	—	—
2.2.3. Are <u>statistics</u> on:		
(i) software code and test errors gathered?	—	—
(ii) software design error gathered?	—	—
2.2.4. Are the following tracked to closure:		
(i) Action items resulting from design/maintenance reviews	—	—
(ii) Action items resulting from code reviews	—	—
(iii) Software trouble reports resulting from testing	—	—
2.2.5. Are there <i>formal</i> organizational <i>procedures</i> for:		
(i) measuring the throughput/effectiveness of the support function?	—	—
(ii) measuring the support staff performance?	—	—
 2.3 Management of the Support Process		
2.3.1. Is a <i>mechanism</i> used for measuring characteristics of the application portfolio (i.e complexity, size, age, technology)?	—	—
2.3.2. Is a <i>mechanism</i> used for measuring and monitoring the costs for each support task/ project?	—	—
2.3.3. Is a <i>mechanism</i> used for measuring and monitoring the workload for each support task/ project?	—	—

Software Supportability Qualitative Assessment Methodology
Organization Assessment Questionnaire

	<u>YES</u>	<u>NO</u>
2.3.4. Is a <i>mechanism</i> used for monitoring and analyzing the nature of each support task performed (nature categorized as perfective, adaptive and corrective)?	—	—
2.3.5. Is there a <i>formal procedure</i> for maintaining support work history?	—	—
2.3.6. Is there a <i>mechanism</i> for tracking the source of maintenance requests?	—	—
2.3.7. Is there a <i>mechanism</i> to ensure that programmers meet scheduled commitments?	—	—
2.3.8. Does senior management have a <i>mechanism</i> for the regular review of the status of software system/units maintained?	—	—
2.3.9. Are the error causes reviewed to determine the process changes required to prevent them?	—	—
2.3.10. Do the technical interchanges include information regarding size, complexity, number of errors, etc.?	—	—
2.3.11. Do you have a <i>formal procedure/schedule</i> for:		
(i) regular technical interchanges with the user?	—	—
(ii) regular technical interchanges with the developer/designer of the system being maintained?	—	—
2.3.12. Is the error data from code reviews and tests analyzed to determine the likely distribution of the errors remaining?	—	—
2.3.13. Do you have a formal procedure for:		
(i) assessing the support process and implementing the recommended improvements (i.e., conducting internal maintenance reviews)?	—	—
(ii) controlling changes to code (software requirements)?	—	—
(iii) deciding when to insert new technology into the support process?	—	—
(iv) managing and supporting the introduction of new technologies?	—	—
(v) recording software unit/system maintenance progress?	—	—
(vi) conducting planned maintenance on application software?	—	—
(vii) conducting periodic maintenance audits?	—	—
2.3.14. Do you have a formal change request review process?	—	—

Software Supportability Qualitative Assessment Methodology
Organization Assessment Questionnaire

	<u>YES</u>	<u>NO</u>
2.3.15. Is there a <i>formal</i> impact analysis procedure adopted in you organization?	___	___
2.3.16. Do you share the results of the impact analysis of the change requested with the user?	___	___
2.3.17. Are the following <u>organizational techniques</u> used:		
(i) change request procedure?	___	___
(ii) scheduled maintenance?	___	___
(iii) formal retest procedures?	___	___
(iv) change request review board?	___	___
(v) chargeback for operations and maintenance?	___	___
(vi) periodic maintenance audit?	___	___
2.3.18. Are the following <u>work methods</u> used:		
(i) top - down design?	___	___
(ii) structured programming?	___	___
(iii) structured walk through?	___	___
(iv) checkpoint review?	___	___
(v) library of previous problems and application functions?	___	___
(vi) test data generator?	___	___
(vii) benchmark testing ?	___	___
(viii) programmer workbench?	___	___
2.3.19 Are the following <u>Monitoring techniques</u> used:		
(i) McCabe's Cyclomatic complexity number?	___	___
(ii) McClure's control flow metric?	___	___
(iii) Henry and Kafara's information flow metric?	___	___

TOOLS AND TECHNOLOGY

3.1 Technology Management:

3.1.1. Is a <i>mechanism</i> used for maintaining awareness of the state-of-the-art in software engineering technology?	___	___
3.1.2. Is a <i>mechanism</i> used for evaluating technologies used by the organization versus those externally available?	___	___

Software Supportability Qualitative Assessment Methodology
Organization Assessment Questionnaire

	<u>YES</u>	<u>NO</u>
3.1.3. Is a <i>mechanism</i> used for deciding when to insert new technology into the maintenance process?	—	—
3.1.4. Is a <i>mechanism</i> used for managing and supporting the introduction of new technologies?	—	—
3.1.5. Is a <i>mechanism</i> used for identifying and replacing obsolete technologies?	—	—
 3.2 Tools and Techology used in the Software Support process:		
3.2.1. Are <u>manual</u> testing techniques used to examine the computer program in order to detect particular characteristics of the code?	—	—
3.2.2. Are <u>automated</u> testing techniques used to examine the computer program in order to detect particular characteristics of the code?	—	—
3.2.3. Which of the following <u>automated testing tools</u> are used:		
(i) <u>Error detection aids</u> - for detecting and removing coding error?	—	—
(ii) <u>Anomaly detection aids</u> - for detecting discrepancies in the form and syntax of the code?	—	—
(iii) <u>Structural analysis</u> - automated techniques for characterizing the logical, data and control structures of computer programs?	—	—
(iv) <u>Test data generation</u> - specially developed to satisfy individual and unique project and test requirements?	—	—
(v) <u>Program metrics</u> - for classifying and estimating error types, performing cost estimations and establishing program complexity measures?	—	—
3.2.4. Are <u>dynamic</u> testing techniques used (which require program execution for analysis)?	—	—
3.2.5. Which of the following <u>dynamic testing tools</u> are used:		
(i) <u>Error detection aids</u> - which involves isolating, detecting and removing errors from program code while executing the program?	—	—
(ii) <u>Structural analysis</u> - as in Automated testing techniques, but requiring program execution for logic, data and code analysis?	—	—
(iii) <u>Functional analysis</u> - black box testing, input/output driven, based on program performance requirements and functionality?	—	—
(iv) <u>Performance monitors</u> - automated techniques to collect performance data on computer program execution characteristics?	—	—

Software Supportability Quantitative Assessment Methodology
Organization Assessment Questionnaire

	YES	NO
3.2.6. Are <u>test management techniques</u> , which aid in the documentation, control and conduct of testing used?	___	___
3.2.7. Are <u>formal techniques</u> involving rigorous symbolic, algebraic or mathematical manipulation of computer programs used for formally verifying computational properties and performance goals?	___	___

Tools/technology issues:

3.3.1. Are the tools used by development compatible with those used by maintenance?	___	___
3.3.2. Do most tools support a majority of the languages being used?	___	___
3.3.3. Do most tools support a majority of the systems being used?	___	___
3.3.4. Is a <i>mechanism</i> used for ensuring smooth transition of a system from development to maintenance?	___	___
3.3.5. Is any effort being made for developing an integrated set of tools to support each phase of the software lifecycle (development through maintenance)?	___	___
3.3.6. Is there a <i>formal process</i> for reporting the size of errors, number of changes made, and complexity of errors to the development group?	___	___

Documentation tools:

3.4. Are the following documentation tools used:		
(i) User manual?	___	___
(ii) Data dictionary?	___	___
(iii) Data flow diagram?	___	___
(iv) Operations error history?	___	___
(v) System maintenance journal?	___	___
(vi) Pseudo-code?	___	___
(vii) HIPO diagram?	___	___
(viii) Data model diagram?	___	___
(ix) Test history?	___	___
(x) Automated code analyzer?	___	___
(xi) Warnier diagram?	___	___
(xii) Jackson diagram?	___	___

Software Supportability Qualitative Assessment Methodology
Organization Assessment Questionnaire

PERSONNEL MANAGEMENT

YES NO

4.1. Training:

4.1.1. Do software personnel simultaneously work on development and maintenance projects?

— —

4.1.2. Is there a *mechanism* for measuring and improving support personnel skills?

— —

4.1.3. Is there a required training program for all newly appointed support managers designed to familiarize them with software support project management?

— —

4.1.4. Is there a required software engineering training program for:

(i) software maintenance personnel?

— —

(ii) first line supervisors of software support?

— —

4.1.5. Is there a formal training program required for:

(i) the support team leaders?

— —

(ii) software change specification writers?

— —

(iii) software test specification writers?

— —

(iv) software documentation writers?

— —

4.1.6. Is there a formal user training program offered for each software product that is being supported?

— —

4.1.7. For those employees with at least 3 years of experience, have at least one third of them had some formal re-currency training in the last 3 years?

— —

4.2 Experience:

4.2.1. Are software personnel rotated between different departments/groups?

— —

4.2.2. Are there several occasions when support personnel are required to work overtime?

— —

4.2.3. Is there a *mechanism* to ensure that there are sufficient knowledgeable and trained support personnel for every application?

— —

Software Supportability Qualitative Assessment Methodology
Organization Assessment Questionnaire

	<u>YES</u>	<u>NO</u>
4.2.4. Do all the software support managers have at least:		
(i) 5 years of experience with the support organization?	—	—
(ii) 10 years of experience with the support organization?	—	—
4.2.5. Do all the software support personnel have at least:		
(i) some formal technical training in the area of software support?	—	—
(ii) a 2 year degree or equivalent in a computer related discipline?	—	—
(iii) a 4 year degree or equivalent?	—	—
(iv) a 4 year degree in computer science or software engineering?	—	—
(v) 3 years experience in software support?	—	—
4.2.6. Do at least two thirds of the software support personnel have:		
(i) at least 6 years experience in software support?	—	—
(ii) at least 1 years experience with all the tools and applications?	—	—
(iii) at least 1 years experience with all the languages used?	—	—
 4.3 Turnover Rate:		
4.3.1. Is the total number of employees that have left the support organization for any reason in the last 3 years:		
(i) less than 10 percent of the current number of support staff?	—	—
(ii) less than 30 percent of the current number of support staff?	—	—
(iii) less than 50 percent of the current number of support staff?	—	—
 4.4 Recruitment/Motivation/Evaluation:		
4.4.1. Is there a <i>formal procedure</i> for:		
(i) career planning?	—	—
(ii) support personnel selection?	—	—
4.4.2. Is there a <i>mechanism</i> for measuring and improving:		
(i) support personnel motivation?	—	—
(ii) support personnel productivity?	—	—
4.4.3. Do you consider the motivation level of staff as reasonably high?	—	—
4.4.4. Do all personnel know exactly what their functions/duties are?	—	—
4.4.5. Would you term the relationship between staff and managers as:		
(i) good?	—	—
(ii) excellent?	—	—

D Matrix

This appendix contains an illustration of the organizational assessment maturity matrix. Maturity levels are depicted on the horizontal axis, and organizational support factors are shown on the vertical axis. Within each matrix cell is a listing of organizational assessment questions corresponding to a given level of maturity and organizational support factor.

Matrix

	LEVEL 1	LEVEL 2	LEVEL 3	LEVEL 4	LEVEL 5
<u>ORG. ISSUES</u>					
1.1.	3	1, 4, 6, 7, 14	2, 8, 9, 12, 13	5, 10, 11	-
1.2.	-	1, 2, 3,	-	-	-
1.3.	-	2	1	-	-
1.4.	-	-	2	3	1
1.5.	3.1	1, 3, 2	2, 3, 3	3, 4	-
1.6.	-	1, 2, 7, 2	3, 4, 7, 2, 8	5, 6, 7, 1, 9	-
1.7.	-	-	-	1, 1, 1, 2	-
<u>S/W PROCESS</u>					
2.1.	-	-	1, 1, 1, 2, 1, 3, 3, 4, 5 7, 8	2, 1, 2, 2, 2, 3, 6	-
2.2.	-	2, 1, 2, 2, 2, 3	4, 1, 4, 2, 4, 3	1, 3, 1, 3, 2, 5, 1, 5, 2	-
2.3.	6, 17, 1	1, 5, 18, 1, 18, 2,	4, 7, 8, 11, 1, 11, 2, 13, 6, 14, 15, 17, 2, 17, 3, 18, 6, 18, 7, 18, 8	2, 3, 9, 10, 12, 13, 2, 13, 4, 13, 5, 13, 7, 16, 17, 5, 17, 6, 17, 4, 19, 1, 19, 2, 19, 3	13, 1, 13, 3
<u>TOOLS/TECH</u>					
3.1.	-	1	-	2, 3, 4, 5	-
3.2.	1, 3, 1, 5, 1	2, 3, 2, 5, 2	3, 3, 3, 4, 4	3, 5, 5, 3, 6	5, 4, 7
3.3.	-	-	1, 2, 3, 4, 6	5	-
3.4.	1, 2	3, 6, 7, 8, 11, 12	4, 5, 9	10	-
<u>PERSONNEL</u>					
4.1.	1	7	2, 4, 5, 6, 1, 6, 2, 6, 3, 6, 4	3	8
4.2.	-	2, 5, 1	1, 4, 1, 5, 5	3, 5, 4, 2, 5, 2, 5, 3, 7, 1, 7, 2, 7, 3	5, 4
4.3.	-	-	3, 3	1, 2	1, 1
4.4.	-	1, 1	1, 2, 5, 1	2, 1, 2, 2, 3, 5, 2	-

E Assessment Recording Form

This appendix contains the form to record the answers to organizational assessment questions (see Appendix C). One copy of this form should be provided along with the questionnaire in Appendix C to each respondent.

Assessment Recording Form

Software Support Organization Assessment

Answers to these questions should reflect standard organizational practice

Question Number	Level	Comments	Select Answer	
1.1.1	2		Y	N
1.1.2	3		Y	N
1.1.3	1		Y	N
1.1.4	2		Y	N
1.1.5	4		Y	N
1.1.6	2		Y	N
1.1.7	2		Y	N
1.1.8	3		Y	N
1.1.9	3		Y	N
1.1.10	4		Y	N
1.1.11	4		Y	N
1.1.12	3		Y	N
1.1.13	3		Y	N
1.1.14	2		Y	N
1.2.1	2		Y	N
1.2.2	2		Y	N
1.2.3	2		Y	N
1.3.1	3		Y	N
1.3.2	2		Y	N
1.4.1	5		Y	N
1.4.2	3		Y	N
1.4.3	4		Y	N
1.5.1	2		Y	N
1.5.2	3		Y	N
1.5.3(i)	1		Y	N
1.5.3(ii)	2		Y	N
1.5.3(iii)	3		Y	N
1.5.3(iv)	4		Y	N
1.6.1	2		Y	N
1.6.2	2		Y	N
1.6.3	3		Y	N
1.6.4	3		Y	N
1.6.5	4		Y	N

Software Support Organization Assessment

Answers to these questions should reflect standard organizational practice

Question Number	Level	Comments	Select Answer	
1.6.6	4		Y	N
1.6.7(i)	4		Y	N
1.6.7(ii)	3		Y	N
1.6.7(iii)	2		Y	N
1.6.8	3		Y	N
1.6.9	4		Y	N
1.7.1(i)	4		Y	N
1.7.1(ii)	4		Y	N
2.1.1(i)	3		Y	N
2.1.1(ii)	3		Y	N
2.1.1(iii)	3		Y	N
2.1.2(i)	4		Y	N
2.1.2(ii)	4		Y	N
2.1.2(iii)	4		Y	N
2.1.3	3		Y	N
2.1.4	3		Y	N
2.1.5	3		Y	N
2.1.6	4		Y	N
2.1.7	3		Y	N
2.1.8	3		Y	N
2.2.1	4		Y	N
2.2.2(i)	2		Y	N
2.2.2(ii)	2		Y	N
2.2.2(iii)	2		Y	N
2.2.3(i)	4		Y	N
2.2.3(ii)	4		Y	N
2.2.4(i)	3		Y	N
2.2.4(ii)	3		Y	N
2.2.4(iii)	3		Y	N
2.2.5(i)	4		Y	N
2.2.5(ii)	4		Y	N
2.3.1	2		Y	N
2.3.2	4		Y	N
2.3.3	4		Y	N

Software Support Organization Assessment

Answers to these questions should reflect standard organizational practice

Question Number	Level	Comments	Select Answer	
2.3.4	3		Y	N
2.3.5	2		Y	N
2.3.6	1		Y	N
2.3.7	3		Y	N
2.3.8	3		Y	N
2.3.9	4		Y	N
2.3.10	4		Y	N
2.3.11(i)	3		Y	N
2.3.11(ii)	3		Y	N
2.3.12	4		Y	N
2.3.13(i)	5		Y	N
2.3.13(ii)	4		Y	N
2.3.13(iii)	5		Y	N
2.3.13(iv)	4		Y	N
2.3.13(v)	4		Y	N
2.3.13(vi)	3		Y	N
2.3.13(vii)	4		Y	N
2.3.14	3		Y	N
2.3.15	3		Y	N
2.3.16	4		Y	N
2.3.17(i)	1		Y	N
2.3.17(ii)	2		Y	N
2.3.17(iii)	3		Y	N
2.3.17(iv)	3		Y	N
2.3.17(v)	4		Y	N
2.3.17(vi)	4		Y	N
2.3.17(vii)	4		Y	N
2.3.18(i)	2		Y	N
2.3.18(ii)	2		Y	N
2.3.18(iii)	4		Y	N
2.3.18(iv)	4		Y	N
2.3.18(v)	4		Y	N
2.3.18(vi)	3		Y	N
2.3.18(vii)	3		Y	N
2.3.18(viii)	3		Y	N

Software Support Organization Assessment

Answers to these questions should reflect standard organizational practice

Question Number	Level	Comments	Select Answer	
2.3.19(i)	4		Y	N
2.3.19(ii)	4		Y	N
2.3.19(iii)	4		Y	N
3.1.1	2		Y	N
3.1.2	4		Y	N
3.1.3	4		Y	N
3.1.4	4		Y	N
3.1.5	4		Y	N
3.2.1	1		Y	N
3.2.2	2		Y	N
3.2.3(i)	1		Y	N
3.2.3(ii)	2		Y	N
3.2.3(iii)	3		Y	N
3.2.3(iv)	3		Y	N
3.2.3(v)	4		Y	N
3.2.4	3		Y	N
3.2.5(i)	1		Y	N
3.2.5(ii)	2		Y	N
3.2.5(iii)	4		Y	N
3.2.5(iv)	5		Y	N
3.2.6	4		Y	N
3.2.7	5		Y	N
3.3.1	3		Y	N
3.3.2	3		Y	N
3.3.3	3		Y	N
3.3.4	3		Y	N
3.3.5	4		Y	N
3.3.6	3		Y	N
3.4(i)	1		Y	N
3.4(ii)	1		Y	N
3.4(iii)	2		Y	N
3.4(iv)	3		Y	N
3.4(v)	3		Y	N
3.4(vi)	2		Y	N

Software Support Organization Assessment

Answers to these questions should reflect standard organizational practice

Question Number	Level	Comments	Select Answer	
3.4(vii)	2		Y	N
3.4(viii)	2		Y	N
3.4(ix)	3		Y	N
3.4(x)	4		Y	N
3.4(xi)	2		Y	N
3.4(xii)	2		Y	N
4.1.1	1		Y	N
4.1.2	3		Y	N
4.1.3	4		Y	N
4.1.4(i)	3		Y	N
4.1.4(ii)	3		Y	N
4.1.5(i)	3		Y	N
4.1.5(ii)	3		Y	N
4.1.5(iii)	3		Y	N
4.1.5(iv)	3		Y	N
4.1.6	2		Y	N
4.1.7	5		Y	N
4.2.1	3		Y	N
4.2.2	2		Y	N
4.2.3	4		Y	N
4.2.4(i)	3		Y	N
4.2.4(ii)	4		Y	N
4.2.5(i)	2		Y	N
4.2.5(ii)	4		Y	N
4.2.5(iii)	4		Y	N
4.2.5(iv)	5		Y	N
4.2.5(v)	3		Y	N
4.2.6(i)	4		Y	N
4.2.6(ii)	4		Y	N
4.2.6(iii)	4		Y	N
4.3.1(i)	5		Y	N
4.3.1(ii)	4		Y	N
4.3.1(iii)	3		Y	N
4.4.1(i)	2		Y	N
4.4.1(ii)	3		Y	N

Software Support Organization Assessment

Answers to these questions should reflect standard organizational practice

[illegible]

F User's Guide

This appendix contains a forms illustrating an 'automated questionnaire' tool that has been developed to facilitate the organizational assessment questionnaire completion process. The appendix also contains an additional "reference questionnaire" illustrating, for each question, the maturity level corresponding to the question.

Software Support Organization Assessment Questionnaire

User's Guide

Users Guide

The automated Software Support Organization Assessment Questionnaire has been designed to facilitate the user in answering the questionnaire by providing a computer guided walkthrough with user-friendly prompts. At the end of a consultation session, the automated questionnaire also calculates the Maturity Level of the organization and indicates areas for improvement.

The automated questionnaire has been developed using VP-Expert, a rule-based expert system development tool for the IBM PC. The automated questionnaire is very easy to use and requires almost no prior PC knowledge.

The following figures give a step by step walkthrough of a consultation session:

USERS GUIDE

STEP 1

Insert the diskette into the A Drive.

At the DOS prompt type VPX

A:\> VPX

The first screen will now appear.

STEP 2

VP - EXPERT Version 2.1							
[RULES]				[FACTS]			
1 Help	2 Induce	3 Edit	4 Consult	5 Tree	6 Filename	7 Path	8 Quit

STEP3

VP - EXPERT
Version 2.1

Choose a file:

QS.KBS

XXXX.KBS

1 Help 2 Induce 3 Edit 4 Consult 5 Tree 6 Filename 7 Path 8 Quit

STEP 4

[KBS.QS]

Loading file...

[RULES]

[FACTS]

1 Help 2 Induce 3 Edit 4 Consult 5 Tree 6 Filename 7 Path 8 Quit

STEP 5

This questionnaire is designed to help determine the software support capability of an organization.

press any key to begin the consultation

STEP 6

SOFTWARE SUPPORT ORGANIZATION ASSESSMENT QUESTIONNAIRE

DEVELOPED BY CIMR

All answers should reflect current organizational practice

<Press any key to Continue>

STEP 7

Are departments, groups in the software organization structured on the basis of life cycle phase (separate Development and Maintenance groups)?

☒ yes

☐ no

Place the cursor on the selected answer by using the arrow keys.

Press "Enter" or "Return".

Another Question Screen similar to the one shown above will appear.

Continue to answer questions from the Question Screens as described above.

Questions with sub-parts will scroll through on the screen. New questions will appear at the top of similar Question Screens.

When all questions in the questionnaire have been answered, the system will compute the organizational level. A flashing screen asking the user to wait will appear. After about 10 seconds, the final screen as shown in STEP 8 will appear.

STEP 8

The Organizational Maturity level is:
LEVEL 2

The areas that need improvement are:

Software Process
Personnel

Press enter or escape to exit.

References

- [CDS86] S. D. Conte, H. E. Dunsmore, and V. Y. Shen. *Software Engineering Metrics and Models*. Benjamin/Cummings, Menlo Park, CA, 1986.
- [DeM82] T. DeMarco. *Controlling Software Projects*. Yourdon, New York, 1982.
- [Dep82] Department of Defense. *Test and Evaluation of System Reliability, Availability, and Maintainability: a Primer*, March 1982. DoD Directive 3235.1-H.
- [IEES3] IEEE. *IEEE Standard Glossary of Software Engineering Terminology*. 1983. Standard 729-1983.
- [PTH87] D. E. Peercy, E. Tomlin, and G. Horlbeck. Assessing software supportability risk: A minitutorial. In *Conference on Software Maintenance*, pages 72-80. IEEE, 1987.
- [Sch90] Stephen R. Schach. *Software Engineering*. Aksen Associates Incorporated Publishers, Boston, MA, 1990.
- [Som89] Ian Sommerville. *Software Engineering*. Addison-Wesley Publishing Company, New York, 1989.
- [War87] J. L. Warthman. Software quality measurement demonstration project (i). Technical Report RADC-TR-87-247, Rome Air Development Center, December 1987.