Software Engineering Institute

A Concept Study for a National Software Engineering Database

P. Van Verth

July 1992

TECHNICAL REPORT
CMU/SEI-92-TR-023

Carnegie Mellon

This technical report was prepared for the

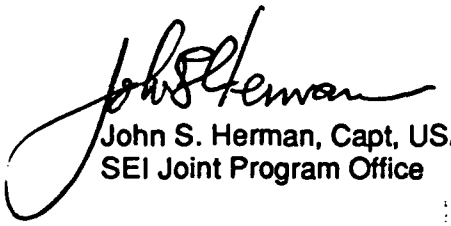
SEI Joint Program Office
ESC/AVS
Hanscom AFB, MA 01731

The ideas and findings in this report should not be construed as an official DoD position. It is published in the interest of scientific and technical information exchange.

Review and Approval

This report has been reviewed and is approved for publication.

FOR THE COMMANDER



John S. Herman, Capt, USAF
SEI Joint Program Office

The Software Engineering Institute is sponsored by the U.S. Department of Defense.
This report was funded by the U.S. Department of Defense.

Table of Contents

Acknowledgments

I would like to gratefully acknowledge the following people for their willingness to be interviewed and their generosity in taking the time to share their thoughts, experiences, and ideas with me. I enjoyed having the opportunity to speak with so many helpful, enthusiastic members of the software engineering community. Several individuals were newly employed at the Software Engineering Institute(SEI) when I interviewed them and I have listed both their previous companies and the SEI as their employers. The following people were interviewed:

William Agresti
The MITRE Corporation

Julia Allen
Software Engineering Institute
Science Applications International Corporation

Henry Apgar
Management Consulting & Research, Inc.

Elizabeth Bailey
Institute for Data Analyses/ Software Metrics, Inc

Bruce Barnes
National Science Foundation

David Card
Computer Sciences Corporation

Andrew Chruscicki
Rome Laboratory

Lori Clarke
University of Massachusetts

Samuel Conte
Purdue University

Barabara Kitchenham
National Computing Centre Limited

Walter Lamia
Software Engineering Institute

Shari Lawrence Pfleeger
The MITRE Corporation

Nat Macon
National Science Foundation

Dale Martin
U.S. Air Force
Space and Missile Systems Center

Michael McCracken
Georgia Institute of Technology

John McGarry
Naval Underwater Systems Center

Frank McGarry
NASA/Goddard Space Flight Center

Celia Modell
Boeing

Barry Corson
U.S. Navy
Naval Air Systems Command

Charles Cox
Naval Warfare Center

Michael Daskalantonakis
Motorola, Inc.

Joseph Dean
Tecolote Research, Inc.

Richard DeMillo
Purdue University

Debbie DeToma
GTE Government Systems Corp.

Erin Dixon
IBM - Federal Systems Company

Peter Dyson
Software Productivity Solutions

Stuart Feldman
Bellcore

Stewart Fenick
U.S. Army
Communications-Electronics Command

William Florac
Software Engineering Institute
IBM Corporation

Peter Freeman
Georgia Institute of Technology

Paul Funch
The MITRE Corporation

John Gaffney
Software Productivity Consortium

Warren Moseley
Texas Instruments

John Musa
AT&T Bell Laboratories

Ron Obranovich
Standards Systems Center

Robert Park
Software Engineering Institute

Robert Paulsen
U.S. Army
Program Manager for Training Devices

Alfred Peschel
TRW

Jerry Pixton
Paramax

Karen Pullen
The MITRE Corporation

Larry Putnam
Quantitative Software Management, Inc.

Samuel Redwine
Software Productivity Consortium

Bernard Roush
NASA
Johnson Space Center

James Rozum
Software Engineering Institute

David Seaver
Project Engineering Inc.

Marie Silverthorn
Texas Instruments

Wolfhart Goethert
Software Engineering Institute
IIT (Illinois Institute of Technology) Research Institute

Robert Grady
Hewlett Packard

John Harding
Bull HN Information Systems, Inc.

Warren Harrison
Portland State University

James Hemsley
Brameur Ltd.

Sallie Henry
Virginia Polytechnic Institute

Watts Humphrey
Software Engineering Institute

Darrel Ince
Open University

Steve Kelly
Kaman Associates

Raymond Kile
Hughes Aircraft Company

Raghu Singh
Space & Naval Warfare Systems Command

Sherry Stukes
Management Consulting & Research, Inc.

Robert Thien
Bellcore

James Tierney
Microsoft Corporation

Jean Tyson
Beckmann Instruments

Peggy Wells
Electronic Systems Center/FMCR

MaryLee Wheaton
Aerospace Corporation

Tony Williams
IIT (Illinois Institute of Technology) Research Institute

Ron Willis
Hughes Aircraft Corporation

Gordon Wright
Naval Oceanic Systems Center

In preparing this report I was aided by the professional support staff of the SEI. Special thanks are owed to Linda Pesante, whose editorial assistance helped shape this document into a final, publishable form, and to Lori Race, who provided outstanding secretarial services when I needed them.

And finally, this report could not have been written without the active participation and contributions from the other members of the SEI Software Process Measurement Project:

John Baumert
Computer Sciences Corporation

Mary Busby
IBM Corporation

Anita Carleton
Software Engineering Institute

William Florac
Software Engineering Institute

Wolfhart Goethert
Software Engineering Institute

Donald McAndrews
Software Engineering Institute

Mark McWhinney
Software Engineering Institute

Robert Park
Software Engineering Institute

James Rozum
Software Engineering Institute

A Concept Study for a National Software Engineering Database

Abstract. Substantial segments of the software engineering community perceive a need for high-quality software engineering data at a national level. A national software engineering database has been proposed as one way to satisfy this need. But is such a database feasible and can it really satisfy these needs?

This report provides information obtained from an informal survey of members of the software engineering community about a national database. The survey served as a means of getting a sense of the expectations that are to be met by building a national database, and provided the opportunity to learn from the experiences of those surveyed about data collection and use. The report summarizes this material in a manner that is informative and expository rather than prescriptive.

1. Introduction

A national software engineering database is a major undertaking that can consume large amounts of money, time, and resources and that requires the cooperation of many elements of the software engineering community. Careful consideration and study are needed before making future decisions about planning or designing a national database. This report is a concept study and a **first** step in exploring the problems and issues surrounding a national software engineering database. The material in this report is based on an informal survey of members of the software engineering community and background information about similar database efforts.

This report lays out preliminary groundwork for future discussions of a national software engineering database. In part it is based on opinions and ideas expressed by representatives from academia, industry, and government. Advice and recommendations from the same sources are included to help those involved make informed decisions. Brief descriptions of similar past and ongoing efforts are included to demonstrate the current state of the practice and provide lessons learned.

This document is not a blueprint for building a national database; it does not contain details about why or how a database should be built. Nor does the existence of this report represent a commitment of the Software Engineering Institute (SEI) to plan or implement a national database. Rather, the report is a loosely structured study with often times conflicting opinions and recommendations from knowledgeable members of the software engineering community. Thus, the objectives of this document are to do the following:

- Articulate needs perceived by the software engineering community.
- Highlight important issues expressed by the software engineering community.
- Present advice and recommendations from the software engineering community.
- Describe briefly past and present similar database efforts.

This report is intended for these readers:

- Anyone seeking quality software engineering data originating from sources other than their own, for example, developers, contractors, acquisition officers, and scientific investigators.
- Anyone, such as program officers, project managers, and senior level management, who might be asked to supply data for a national database.
- Anyone with an interest in building a national software engineering database.

1.1. Organization of the Report

The report has two components: 1) a brief background study of database efforts with characteristics similar to a national database, 2) a summary of results of a survey of the software community about a national database. Chapter 3 presents a capsule view of four of these databases; material in the other chapters also contains example practices and lessons learned from these efforts.

Survey results make up the major portion of the report. Result categories include descriptions of what people think a national database should be (Chapter 4), problems they foresee will affect implementation (Chapter 5), reservations they have about the idea of national database (Chapter 6), and advice for those involved with any such effort (Chapter 7).

Chapter 8 summarizes the conclusions derived from the study.

1.2. Role of a Database in Software Measurement Activities

Measurement provides a quantitative basis for assessing software development processes and products. Measurement is taking on an increasingly important role in software development. Measurement programs are being initiated not only at the project or company level but also at the national level as well. National goals are being set for the software community that rely upon measurement to determine whether the goals have been met. For example, the DoD Software Technology Strategy [DoD 91] has three objectives to be achieved by the year 2000:

- Reduce equivalent software life-cycle costs by a factor of two.
- Reduce software problem rates by a factor of ten.
- Achieve new levels of DoD mission capability and interoperability via software.

A necessary component of any software measurement effort, including a national one, is a database in which to record and analyze measurement data. Databases play important roles in measurement efforts throughout the software community. Example databases include the Software Engineering Laboratory (SEL) [Valett 89], Hewlett-Packard [Grady 87], the Software Engineering Research Center (SERC) [Yu 88], the Automated Measurement System (AMS) [RADC 88], the Software Data Library (SWDL) [SWDL 87], and the Software Evaluation and

Testing Panel (STEP) [Beavers 91]. Moreover, these examples show that before establishing a database within a measurement program, measures and data collecting methods need to be defined.

Data from companies differs widely between organizations because the software community has no standard software measures or measurement practices. Uses of data from a national database, e.g., to make comparisons, to do statistical analysis, or to examine historical trends, require data that is counted and measured consistently. Demands for consistent data across organizations have spurred various national and international groups to sponsor efforts to formulate common definitions and measurement standards. These groups include the IEEE [IEEE 90] among others. Common definitions ensure comparability of data and communicate unambiguous meaning. Common definitions lead to standardized collection methods and eventually to the possible realization of a national database.

1.3 Overview of Survey Results

Common goals appear to underlie many of the opinions, concerns, and comments of those interviewed about a database. These goals were not addressed directly in the interviews; nevertheless, they may be summarized this way:

- Improve the way software is being developed.
- Understand the software process better.
- Make a better product.
- Gain or maintain a competitive edge.

Survey results are better understood when seen in the context of these more general goals. The following observations summarize survey results:

1. Needs - The most frequently cited need was for data to compare "my" organization with others.
2. Readiness - The software community appears ready for a national database; however, there was concern about the overall maturity of the community in being able to use the data judiciously.
3. Data Collection - Data collection practices and reasons for collecting data vary too much between organizations at this time to make a database feasible.
3. Data Sources - There are relatively few organizations in a position to supply data for a database in contrast to the number of organizations that are ready to use the data.
4. Getting Started - Initial efforts should be small, simple, and well-planned in advance.
5. Critical Issues - The key issues for the success of a database are having common data definitions and protecting the confidentiality of the data.

The remainder of the report elaborates on these points.

2. Approach to the Study

Investigating similar databases provided a way of obtaining lessons learned from previous and ongoing database efforts, and a way of assessing the current state of the practice. This report does not describe all applicable databases; examples were selected by length of experience, noted success, or depth of planning and design.

The informal interviews and conversations making up the survey component of the study are the basis for the material in Chapters 4 - 7. Personal or telephone interviews were selected as the best method for gathering the open-ended kinds of information desired (opinions, experiences, and ideas). Interviews covered three main topics:

- What did interviewees have in mind when thinking about a national database.
- What concerns and issues did interviewees have about providing data for or using data from a database.
- What experiences, advice, and historical perspectives did interviewees have to offer about a national database.

The 10 month time period limited the kinds of approaches. The informal personal interview was chosen over a formal survey questionnaire because formal techniques require more time than was available to prepare, conduct the survey, and analyze the results. Moreover, at this early stage, gathering opinions, experiences, and observations was more important than obtaining numerical data for statistical analysis.

The interviews revolved around a basic set of questions. These questions covered general items such as the purposes of a database, data items to be collected, and questions about concerns and advice. The basis for some of the interview questions was the Goal-Question-Measure (GQM) approach advocated by Basili [Basili 85]. Putting together a national database has similar characteristics to setting up a software measurement program for which GQM is intended. GQM as a framework for planning a software engineering database helps keep the planning and design process focused. In reporting interview results, one topic, purposes, included questions about needs and goals because many responses blurred any distinctions between the two.

Interviews varied in the number of "canned" questions that were actually asked. The length of time for an interview also varied and depended upon the depth of experience of the participant and willingness to share information. Additionally, if a participant had ideas and experiences in a particular domain, the basic set of questions was supplemented to pursue to this new area of interest. Thus, the interviews tended to be casual, informal, and exploratory.

A variety of sources provided the names of those interviewed. Names included members of the SEI Software Process Measurement Group, members of the Measurement Steering Committee, and members of the various measurement working groups. Conversations with participants produced follow-up names. Participants were also selected so that the academic, industrial, and government communities had fair representation. The list does not

include everyone with valuable information to contribute. It should be broad enough to provide a sense of what opinions might prevail in all communities. Appendix B provides a sample interview form.

2.1. Business of Supporting Organizations

In keeping with the SEI commitment to "establishing, promoting, and participating in cooperative software related efforts among industry, government, and academia," each of these communities had a representative sample. There were 7 researchers from academia, 46 practitioners and researchers from industry, and 13 practitioners and researchers from government. The academics hold positions at universities specializing in software engineering research or closely related fields. The industrial category included DoD-related developers and contractors and developers from the commercial sector as well as federally funded research and development centers (FFRDCs) like the SEI, the MITRE Corporation, and the Institute for Defense Analyses (IDA). The government category included government program offices distributed among the services.

2.2. Description of Interview Participants

Participant interviews used the following three operational perspectives on a database:

- Data users - use data from the database.
- Data suppliers - collect raw data and turn it over to the data collectors.
- Data collectors - accept the raw data from the suppliers and prepare it for entry into the database.

Participants could fill all three roles, and some did.

2.2.1. Data Users

Interviews started with questions about using a national database, for example, needs, goals, etc. for a national database. Several people interviewed did express reservations about the value of "national" data. A skepticism about the success of the database effort, the quality of the data in such a database, and the ability to translate data to local contexts were the bases of these reservations. More typically, the majority of those interviewed had needs for data that could not be satisfied within their own organizations. Some of those needs were simple, such as wanting to see how others were doing in comparison to their own organizations. Others needs were more profound and reflected an organization's inability to generate data altogether, e.g., academic researchers. User needs also ranged over a wide variety of data objects, from actual software artifacts to process asset libraries to various kinds of process and product metrics. Needs also differed depending upon the viewpoint of the representative organization. For example, some program acquisition officers wanted schedule and effort

data while some contractors were looking for data that would give them insight into new technologies and methods. Chapter 4 discusses additional details about users needs.

2.2.2. Data Suppliers

Data suppliers are those whose organizations, mainly developers and contractors, have the potential to be a source of data for the database. Most organizations do some data collecting and it is from these collections that data for a national database will most likely be taken. Variations on the goals and objectives of a software engineering database may require data collection targeted especially for the database; however, early database designs will probably rely on existing data. Interview participants identified as data suppliers were not asked for donations of organizational data, nor were they asked for any future commitments to supply data. Chapters 5 and 6 discuss many of the issues and concerns of data suppliers.

2.2.3. Data Collectors

Data collectors are those responsible for or experienced in handling data collected from multiple sources (not just their own local context). They have knowledge and expertise in matters relating to establishing, maintaining, and administrating a data collection or database of software development data. Many of those interviewed in this category were associated with program offices, consortia, specially organized offices or agencies, for example, Space Systems Cost Analysis Group (SSCAG), Space and Missile Systems Center (SMC), the Software Engineering Laboratory (SEL), the Data and Analysis Center for Software (DACS), Electronic Systems Center (ESC), and the MITRE Corporation. Their special insights into the issues about software engineering databases provided material for Chapter 5.

3. Existing Databases

There are several software engineering databases that serve as examples and models, and provide historical perspectives for a national database effort. Three of the databases are operational, the Data and Analysis Center (DACS) Software Life Cycle Experience Database (SLED), Space and Missile Systems Center (SMC) Software Development Database (SDDB), the Software Engineering Laboratory (SEL) Data Base. The fourth, the Software Data Library (SWDL), never fully achieved its goals but is notable for its planning and design. This chapter gives a capsule summary of each database.

3.1. Data and Analysis Center for Software (DACS) - Software Life Cycle Experience Database (SLED)

When software engineering databases are mentioned, DACS SLED databases are those that come immediately to mind. The DACS has been in existence for a long time and was one of the first organizations to recognize the need for software engineering data on a national scale. DACS is a DoD Information Analysis Center sponsored by the Defense Logistics Agency (DLA) and was established "to provide a focal point for software development and experience data and information within the field of software engineering" [DACS 91]. The DACS SLED has a number of data collections that it has obtained from other sources, e.g., the SEL, AT&T, ESD, in addition to data collections that it has formed on its own.

Early DACS data collection efforts were limited by the fact that the data sets were supplied on voluntary basis. DACS had no money or authority to ensure the consistency and validity of the data and accepted whatever data was offered. As a consequence data in the early DACS collections was not as useful as some had hoped it would be. There were problems with incompatible formats, differing levels of detail, and differing counting methods for collected data that made use of the data for comparison purposes difficult. These early experiences of DACS have had strong influences on later database efforts both at DACS and elsewhere in avoiding similar pitfalls.

3.1.1. Details

Name:

- Data Analysis Center for Software (DACS) Software Life Cycle Experience Database (SLED)

Sponsor:

- Defense Logistics Agency (DLA), Defense Technical Information Center, (DTIC) Air Force Rome Laboratory (RL)

Contractor Organization:

- Data Analysis Center for Software

Date Started:

- 1978

Goals:

- Support software technology research.
- Assist in the transition and application of new software technology.
- Serve as a rapid, authoritative source of information concerning software technology.
- Function as a repository for information and data related to software engineering.

Types of Data - The SLED database consists of 9 separate data sets:

- Architecture Research Facility Error Dataset
- DACS Data Compendium
- DACS Productivity Dataset
- NASA/Ames Dataset
- NASA/SEL Data Compendium
- NASA/SEL Dataset (mentioned in 3.2)
- Phased Array Warning Systems (PAVE) PAWS Operations and Maintenance (O&M) Dataset
- Software Reliability Dataset
- Validation and Verification Dataset

Sources of Data:

- Various organizations

Uses of Database:

- Software cost, reliability, and quality research

Number of Sample Points:

- 9 datasets, each with multiple sample points

Distribution of Data:

- Available to the public on magnetic tape or in hard copy

Services Provided to Users:

- Range of services in combination with other DACS products

3.2. Software Engineering Laboratory (SEL) Data Base

The SEL Data Base is the product of many years of collaboration in software engineering research between the National Aeronautics and Space Administration /Goddard Space Flight Center (NASA/GFSC), the Computer Science Department of the University of Maryland, and Computer Sciences Corporation. The SEL Data Base contains data collected from specially

selected flight dynamic projects targeted for research purposes [Valett 89]. SEL data is limited in scope and general applicability since the target projects themselves are so focused. The SEL Data Base is an example of a database and data collection effort that is highly successful primarily because it has well-defined objectives and has been carefully planned. Although data may not be useful outside the SEL environment, it is available to the public through the Data and Analysis Center in Utica, New York. The SEL has many publications relating the results of their studies [SEL 89].

Lessons Learned:

In an early paper, Basili reported many lessons learned [Basili 1984]:

- Understand the work environment and the people involved. Train staff to prevent misunderstandings about why data is being collected and provide feedback to them after collection is completed.
- Validate data in a timely fashion so problems can be detected and corrected early in the collection process.
- Minimize overhead in the collection process itself, that is, keep manual forms short and to the point. Automate where possible.
- Check data for consistency and accuracy before analyzing it.
- Understand factors affecting the data before interpreting results.
- Allocate sufficient resources for verification and validation. It is easy to underestimate the resources needed for checking the quality of data.
- Be aware of the sensitivity of data and protect the data accordingly.
- Keep initial data collection efforts small.

In a later paper, Valett and McGarry reported additional lessons learned [Valett 89]:

- Not all data is useful; collecting too much data can be a problem. SEL classifies data as: critical data, data useful to specific studies, data of little or no use. Critical data is of importance to an organization under all circumstances and should always be collected. and critical data is generally inexpensive to collect. For SEL this includes information on phase dates, resource data, and a record of changes and errors. Data useful to specific studies is collected on an as-needed basis and may be more expensive to collect than critical data. Examples of this kind of data in the SEL are detailed error data and effort data by component. Least-used data is that which SEL has found to be least useful although collected in the past.
- An overhead of about 5% was incurred on early projects. Later projects incurred costs of 2-3% of the build cost. SEL data processing costs after initial collection are an additional 5% of development effort. This includes validation and verification, entry into the database, and archiving. Staff hours including initial start-up costs are estimated at 3-10 staff-years of effort.

3.2.1. Details

Name:

- Software Engineering Laboratory (SEL) Data Base

Sponsor:

- National Aeronautics and Space Administration/Goddard Space Flight Center (NASA/GSFC)

Organizational Members:

- NASA/GSFC, Software Engineering Branch
- The University of Maryland, Computer Science Department
- Computer Sciences Corporation, Systems Development Operation

Date Started:

- 1976

Goals:

- Understand the software development process in the GSFC environment
- Measure the effects of various methodologies, tools, and models on this process
- Identify and apply successful development practices

Typical Data:

- Resource data
- Error data
- Product characteristics
- Estimates history
- Growth history
- Change history
- Project characteristics

Sources of Data:

- Member organizations of SEL

Uses of Database:

- Research

Number of Sample Points:

- April, 1991 - data from 104 flight dynamics projects

Distribution of Data:

- Public access through DACS

Services Provided to Users:

- None to the general public

3.3. Software Development Database (SDDB)

The SDDB is an outgrowth of the Space System Cost Analysis Group (SSCAG) database organized by the Software Subgroup. SSCAG is a government/industry group formed to provide a forum where everyone concerned with performance and cost in high-technology space systems could state views, exchange information, and obtain acceptable compromises. A subset of this group has contributed to the SSCAG database. The Air Force Space and Missile Systems Center (SMC) oversees and supports the SDDB. Management Consulting and Research, Inc. (MCR), a contractor that specializes in cost analysis support for the database, receives all data from contributors, verifies and sanitizes the data, enters it into the database, and then distributes copies of the database per Air Force instruction. Distribution is limited to only those organizations that have contributed data or have been cleared by the SMC. SDDB is implemented in dBase IV and copies are distributed on diskette in dBase IV format. MCR receives data from suppliers in either electronic form on diskette or in hard copy. MCR trains data providers on request in the use of the database and in data collection methods. Data providers are relied upon to provide good quality data; there are no formal validation and verification procedures. Data providers may request on-site assistance from MCR in preparing and submitting data. Users of the database (in this case the same group as the suppliers) use the data for research and for internal purposes related to cost estimation and planning. There have been several publications resulting from the SDDB data. Plans are in progress to expand the database to include data from the maintenance phase of the software process.

The SDDB has reported the following lessons learned [private communication]:

- Ensure Database management software will meet user needs.
- Determine the application of data before collecting it.
- Commercial cost models require user calibration.
- Software calibration is unique to the environment.

3.3.1. Details

Name:

- Software Development Database (SDDB) formerly the Space System Cost Analysis Group (SSCAG) Data Base

Sponsor:

- USAF Space and Missile System Center

Contractor Organization:

- Management Consulting and Research, Inc.

Date Started:

- 1983 - original SSCAG Data Base

Goals:

- Support accurate software program sizing and cost estimating

Types of Data :

- Software cost, size, and schedule data with attributes for several cost models - mainly from space-related software development projects

Data Sources:

- Contributions from members of SSCAG - government and industry
- Portions of SMC and Electronic Systems Center (ESC) software sizing databases

Uses of Database:

- Parametrics estimation
- Model calibration
- Estimation by analogy
- Estimate validation

Number of Sample Points:

- 2, 545 at the project and CSCI level

Distribution of Data:

- Sanitized versions of data in dBase IV format to all government agencies and other SSCAG data contributors on a restricted basis

Services Provided to Users:

- Software for data analysis
- Disk copy of database (assumes recipient has a copy of dBase IV)
- Documentation of the data collection, normalization techniques, etc.

Documentation:

- There are no public documents for SSDB but Captain Dale Martin, SMC, El Segundo, California, will provide complete documentation to government agencies and organizations contributing data to the effort.

3.4. Software Data Library (SWDL)

The SWDL project was set up to plan and implement a national database of software data in the United Kingdom during the 1980s. It was sponsored by the United Kingdom Alvey Project and was developed by a consortium of five organizations. The project entailed three phases, 1) a feasibility study, 2) implementation, 3) self-support. Phases 1 and 2 of the project were realized and the project terminated in 1988 before initiating phase 3. [SWDL 87]. The project was considered a partial success.

It was successful in doing the following [SWDL 89]:

- Developing a data model and data definitions for a metrics database.
- Implementing a model database.
- Producing guidelines for data collection and data analysis.
- Publicizing concepts and procedures for doing software metrics.
- Collecting and analyzing data from a number of software projects.

Lessons that were learned from the project [SWDL 89]:

- The attempt to support both industry needs and research needs led to a overly-complex data model that was unsatisfactory for the needs of both communities.
- The data model was made even more complex by trying to ensure comparability of data from different projects without common data definitions.
- There was confusion over who should actually collect data for the database, the consortium or the participating companies. The companies expected the consortium to collect data and the consortium did not have enough personnel for the task. The consortium had hoped to encourage companies to start their own metrics programs.

3.4.1. Details

Name:

- Software Data Library (SWDL)

Sponsor:

- United Kingdom Alvey Project

Contractor Organizations:

- The National Centre for Systems Reliability
- International Computers Limited
- The National Computing Centre Limited
- GEC Software
- Logica UK

Date Started:

- 1984

Goals:

- Provide information on industrial software development
- Provide a database of performance, reliability, and cost metrics

Types of Data:

- Change
- Resource
- Personnel

- Product
- Development environment
- Operational environment

Data Sources:

- Various UK companies

Distribution of Data:

- On-line access through specialized interfaces

Services Provided to Users:

- Data analysis
- Statistical techniques
- Data presentation techniques (text and graphics)

4. Descriptions of a National Database

Chapter 4 presents interviewees' descriptions of a national software engineering database, the purposes it should satisfy, questions it should answer, and data that should be collected. This chapter discusses the conceptual aspects of the database; chapter 5 presents details and implementation issues. The surveys provided material for the chapter, and clarifications have been added when needed.

Interviewees described four types of national databases, each serving different purposes:

- A macro-level database that focuses on the U.S. software industry as a whole. This type of database encompasses the whole of the industry, summarizing U.S. trends and progress in software development at a very high and general level. This is truly a "national" database.
- A micro-level database that focuses on the U.S. software industry at the organizational or project level. This type of database has an organizational focus, data in the database is more detailed, possibly broken down to the level of individual projects or subunits, and allows individual organizations to use the data for comparisons or benchmarking.
- A database that supports new technologies and methodologies. This type of database bridges the gap between the research database and the macro- and micro-level databases with data about the infusion of new technologies and methodologies into the software development process.
- A database for research. This type of database supplies data for researchers to investigate fundamental questions about software engineering. An artifacts database supporting software artifacts research is an example of a research database.

The remainder of this chapter takes each database description and looks at what interview participants had to suggest for purposes (needs and goals), questions to be answered by the data, and data items to be collected.

4.1. National Database - Macro Level

Several interview participants used the term "national database" to describe a database that reflects national (U.S.) software industry goals and needs. They envisioned this database at a macro-level and one that summarizes the state of the U.S. software industry as a whole. This kind of database was viewed by some as a first step toward the more detailed or micro-level database section 4.2 describes.

4.1.1. Purposes

Participants identified several purposes for a macro-level database. For them, this kind of database gives a sense of how the U.S. software industry is doing overall and serves the following purposes:

- Establish baselines for the U.S. software industry.
- Perform studies on the U.S. software industry.
- Track software-related resources and artifacts in the U.S. software inventory.

Establish baselines for the U.S. software industry. Participants emphasized the importance of establishing industry-wide baselines to provide a measure of where the industry is currently positioned. Baselines characterize software and software processes for later trend analysis and comparisons. Examples given of attributes to baseline include Cost/SLOC, SLOC/Software Unit, or average cost to develop large-scale systems. Baseline data is also used to track information within the various software engineering communities, for example, commercial or government sector, or within application domains, such as avionics, aerospace, operating systems, banking, and mission critical software.

Perform studies on the U.S. software industry. Interviewees thought that a database with information about the U.S. software industry can be used in studies appraising the strengths and weaknesses of the industry. Example studies include trend and growth analyses of the size and complexity of software systems, productivity of U.S. programmers, or the quality of U.S. software. Macro-level data supports comparative studies also, e.g., comparing U.S. software industry and the software industries of other nations. One interviewee proposed that quality data can be used to compare the quality of U.S. software to the quality of Japanese software. National productivity is another performance measure of interest.

Track U.S. software-related resources. Interviewees suggested using the macro-level database to track software-related resources in the U.S. to anticipate future needs for those resources. For example, increases in national demands for software and software systems may require sufficient resources to meet those demands, i.e., will there be enough software engineers, technicians, available computer time, hardware, etc.? There is also a need to provide visibility into the dependency of the nation on software, to show the importance of software in critical areas and where it is most likely to affect national interests.

4.1.2. Example Questions

The following questions were suggested by interviewees as the type of questions macro-level data helps answer. Many of these questions may already be answered by existing sources of data; however, they are the kinds of questions that participants expected a macro-level database to answer readily. Participants suggested these questions:

- How much software is there in the nation's inventory?
- How much of the national budget is being spent on software development?
- What percentage of the nation's work force is involved in software production?

- Is the size of software systems getting larger? smaller? by what percentage amount?
- What is the average size of an operating system? banking application? inertial guidance system?
- What is the overall quality of U.S. software?
- What's the percentage of reuse in U.S. software development?
- How many developers are using CASE tools? C++? Object-oriented design?
- What percentage of the software industry is developing avionics systems, commercial applications, mission critical software, etc.?
- Are there sufficient programmers to meet national needs? available computer time?
- How is the U.S. doing compared to Japan with respect to productivity?

4.1.3. Kinds of Data

Interviewees suggested that data collected at the macro-level include project post-mortems and wrap-ups with figures for cost, effort, schedule, size and quality. According to some of the participants, this kind of data is not difficult to obtain since it is data normally provided to customers and program offices by software developers and contractors. Interviewees felt that this kind of data was largely available in the public record, e.g. GAO, National Research Council, or national software industrial publications. An advantage to data at a level this high is that individual organizations do not have to be relied upon directly to supply it. Obtaining data is a matter of locating the relevant independent sources and recording the information.

4.2. A National Database - Micro Level

A national database with a focus on companies and projects provides the software engineering community with more detailed data than that available at the macro level. Detailed data includes company and project attributes, descriptions of the application domain, or the development environment. At this level of detail the cooperation of developers and contractors is needed to collect and supply data. Interviewees expressed serious reservations about this kind of database, especially about the quality and comparability of the data, and the ability of organizations to use the data effectively. This chapter presents participants ideas about this type of database while Chapter 6 has more details about their concerns.

4.2.1. Purposes

There was a strong response during the interviews about developer and contractor needs for data at the organizational or project level. Interviewees wanted data to see how their organizations were doing with respect to other organizations or with respect to the rest of the software industry in general. Some wanted data to do "sanity checks" with their own

observations and calculations or provide more confidence in their own internal estimates and planning. Most expressed the need for data with enough detail and attribute information to be able to determine if the data had relevance for a particular use or within a given context.

Participants wanted micro-level data to do the following:

- Make process improvements.
- Set benchmarks.
- Develop industry standards.
- Make comparisons.
- Do "sanity checks".
- Analyze risk.

Set benchmarks and develop industry standards

Benchmarking in software engineering implies identifying those attributes to which to apply standards, and then determining the values which are "best" either by defining a value or by observing the best value in software engineering practice. Benchmarking enables the best in the industry, the best in category, and the best practices to be highlighted. Interviewees mentioned benchmarking as a possible prelude to formulating a Malcolm Baldrige Award for software development. Developing industry standards follows naturally from benchmarking.

Quality, productivity, reliability, customer satisfaction, and on-time delivery are candidate attributes for benchmarking. With a set of national benchmarks, a software developer can make a self-evaluation with respect to the rest of the industry to determine whether individual performance is above or below standard and by how much. Benchmarking helps in improving the state of the practice by identifying methods and methodologies that produce the benchmark values. Other segments of American industry such as the American Automotive Industry, American Refrigerator Manufacturers use benchmarks.

Make comparisons

Making comparisons to national performance averages is a purpose similar to benchmarking but with a slightly different emphasis. Performance averages represent the state of the current practice, not the best in current practice. For example, a benchmark for quality might be 5 errors/1000 SLOC while a performance average might be 50 errors/1000 SLOC. Performance averages are obtained by taking the statistical mean of a set of sample points while benchmarks are obtained by defining or observing the best in performance (the highest, lowest, fewest, etc.).

Do "sanity checks"

"Sanity checks" are a way of using performance averages to evaluate or check internal software development processes. For example, an internal estimate to produce a new operating system might be 2 calendar years of time, $5,000,000 in cost. A "sanity check" against a micro-level database might show this estimate to be too low, just about right, or too

high. Estimators are then able to go back and determine whether their figures are justified in light of this information and make adjustments if necessary.

Analyze risk

Risk analysis enables both sponsors and contractors to determine what the risks might be if a contract is awarded or a bid submitted for a software system. Micro-level data facilitates checking to see if company profiles, plans, and estimates are in line with summary descriptions of similar efforts in the past. Note that this is NOT the same as doing company-to-company comparisons or evaluations. Risk analysis in this sense means comparing a single company to summary data, that is, averages, trends, etc.

4.2.2. Example Questions

The following questions may be answered by existing data sources; however, participants thought they should be specifically addressed by a micro-level database. Participants suggested the following questions:

- What are the best doing?
- What are the performance averages for quality in operating systems?
- How long do projects like this take?
- How well/poorly is my project/organization doing with respect to the benchmarks?
- How much has it cost to build systems of this type in the past?
- Am I in trouble?
- What is a good ball park figure for size in projects of this type? cost? effort?
- How many projects like this have been attempted? Were they successful?
- Does my company have what it takes to bid on this system?
- Is this a reasonable proposal?
- What percentage of effort is spent in design, coding, testing, etc.?
- At what milestones should I be 20% into costs?
- Is the cost for this proposal reasonable?
- How do people design? code? test?
- What are the methods that work best?
- How can I do things quicker? better? with fewer people?
- What do I need to do to reduce my risks?

4.2.3. Kinds of Data

Size, cost, effort, schedule, and quality measures are among those data items most frequently mentioned in the interviews. Many of those surveyed wanted data described to

enough detail so that honest, meaningful comparisons could be made, for example, avionics systems of 1 million SLOC compared to similar systems.

The participants suggested further details for data items:

- Size data includes lines of code, function points, number of modules, and subsystems. Refine size by reused, off-the-shelf, new, modified code.
- Cost, effort, and schedule data needs to be supported by information on cost drivers for a variety of cost models. These include resource data, environment data, staffing data, information about the application and its complexity, and phase and activity information.
- Quality data includes errors, problem reports, and defects.

Recommendations for additional data items included the number of requirements, requirements volatility, Capability Maturity Model (CMM) maturity level, customer profiles, project constraints, and information about whether projects met budgets and schedules, and amounts by which they did not. Others suggested that planned as well as actual data is useful. Data items at several levels of detail were proposed: project level or subsystem level, CSCI level or component level, or data at the module or work breakdown structure level. From their replies it was apparent that most participants assumed that all data would be collected after a project or build had been completed, and that data for work in progress would not be collected nationally (although it would have to be collected locally as work progressed).

The survey did not ask for explicit details or definitions of data items; however, most participants were concerned that standardized definitions be used to ensure that data is comparable and meaningful. Another important consideration in populating the database, which participants cited, is the ability to verify and validate data. Chapter 5 contains further discussion of these implementations issues.

4.3. Database Supporting New Technologies and Methodologies

There were frequent calls by interviewees for data to support new technologies and methodologies. This kind of data shows, for example, the cost, impact, and lead time to incorporate new technologies. Examples of new technologies and methodologies include the use of Ada, CASE tools, or object-oriented design. For the purposes of this report new technologies and methodologies also include process improvement techniques. There are few sample points for this kind of database as compared to the number of sample points populating the micro-level database.

4.3.1. Purposes

Interviewees proposed these uses for the database:

- Show the impact of new technologies.

- Provide information so that people will know who is doing what, can call to learn, and can find the right expert.
- House a component library, a reference source for the best practices, handbooks, project histories, successes, failures, and lessons learned.
- Reference a repository of successful applications, an asset library.
- Provide process improvement data, including how the improvement was accomplished, how much it cost, and how long it took.

A database set up for the purposes of evaluating new technologies and methodologies enables others in the software industry to see the impact and cost-effectiveness of a specific technology or methodology on software development. Survey participants considered this an important goal in helping their organizations bring about technological change and reduce some of the risks of trying something new.

The remaining purposes stated above reflect the needs perceived by some to consolidate what is known about new technologies, process improvement, software development in a central, easy to access repository or library. There seemed to be a frustration with the current methods of disseminating information about making improvements and locating sources of expertise and help.

4.3.2. Example Questions

- Does this technology really improve quality? productivity?
- What does it cost to move up a CMM level? How long does it take?
- What is the cost-effectiveness of using the XYZ model/metric?
- What is the value-added for quality, productivity?
- What is needed to improve my process?
- What is needed to reduce the number of errors?
- How does the XYZ model fit my process?
- What is the extent of the impact of X?
- How does new technology affect process?
- How much does it cost to get a pay off?
- How long does it take to get a pay off ?
- Who can help me do Y?

4.3.3. Kinds of Data

Participants suggested the following data items to be collected for supporting new technologies:

- Size, cost, schedule, effort, quality data and any other data that shows the effect of introducing the new technology
- Software process assessment data
- Software capability evaluation data
- Standards, policies, practices
- Tools
- Lessons learned
- Training materials
- Bibliographies, reference materials

4.4. Research Databases

A national database for software engineering research is one in which data is made available for research purposes and where research results are documented. Two types of research databases were mentioned in the interviews: research on models and software artifact research.

4.4.1. A Database for Research on Software Engineering Models

Software model research includes research on models of the software development process and resulting products. Model research includes size, cost, schedule, and effort models, as well as those for quality, reliability, complexity, etc. Model research also includes the formulation and testing of new models.

4.4.1.1. Purposes

Purposes expressed in the interviews provide data to do the following:

- Investigate values for parametric models and the effects of various attributes on models, for example, experience level, environment, and application domain.
- Investigate the relationship between actual values and estimated or predicted values for predictive models, e.g., cost and schedule models.
- Validate models and test model performance on more than limited or selected data sets.
- Have the current reported best values of various parameters for the different cost models (COCOMO, Price-S, SLIM, REVIC, SEER etc.) and performance characteristics of each.

- Conduct research on quality by looking at performance, estimation, and prediction of quality.
- Test and validate reliability models.

An important purpose in model research is the ability to have public, standardized data sets for numerous models that will allow researchers to duplicate experiments and compare performance of new models with old ones.

4.4.1.2. Example Questions

- What parameters are best for model X in application Y?
- Does model X work in environment Z?
- How does model R compare to model S in Ada applications?
- How well does model Q predict errors?
- Does this new model perform better on this data set than model M?

4.4.1.3. Kinds of Data

Data for model research includes cost, schedule, effort, and size with enough details to support most models. Information characterizing the project from which data has been collected includes hardware, attributes such as experience level of staff, familiarity with application, context information, activity, and phase information. Additionally some of those surveyed thought that CMM maturity level, use of specific technologies, and whether a project was over-budget or over-schedule was useful information. Planned versus actual information was also requested. Data in this category differs little from the data described in section 4.2.3. except that it is intended for research on a specific model or models and is thereby more focused. Sources for model data are developers and contractors doing software development and who are using cost models in planning and estimation.

4.4.2. Artifacts Database

An artifacts database is the most open-ended type of database of those described in this report. It is the hardest to satisfy because of this broad scope. Donald Knuth's study of programmer coding patterns in compiler optimization techniques is an early example of software artifact research [Knuth 71]. Models for artifact research exist in other engineering disciplines such as aerospace engineering or automotive engineering. The research community considers software engineering artifact research important enough that the National Science Foundation sponsored a workshop in January 1990 evaluating the need for artifact research.

4.4.2.1. Purposes

Most of the academic researchers interviewed had particular interest in a software engineering artifact database. The participants cited the following uses for such a database:

- Characterizing large scale software systems.
- Investigating relationships between process and product attributes.
- Validating theories and models about software development.
- Examining product characteristics.

4.4.2.2. Example Questions

The following are only a few of the basic research questions that having an artifact database will help answer:

- What characterizes large-scale systems?
- What effects does a programming language have on software quality?
- What effects do programming language features have on maintainability? testability? complexity? reliability?
- What are the structures underlying software systems?

4.4.2.3. Kinds of Data

Artifact data includes all the tangible products resulting from the development of software, for example, source code, design documents, user manuals, problem reports, testing plans, and testing results. Having artifact data gives researchers the ability to go back to original sources to verify what is there, check definitions, ensure consistency, etc. Many researchers felt that data collection during the development process was undisciplined and inconsistent as currently practiced. An artifact database allows researchers to look for the microscopic details that are possibly missing from data collected during the development process.

5. Issues

Planning and designing a database involves many decisions about a variety of factors, e.g., logical and physical structure of the database, database administration, database services and maintenance. During the interviews, participants brought up issues they felt were particularly critical to establishing a national software engineering database. These include: 1) implementation issues about data definitions, confidentiality, supporting information, validation and verification, adding value, and access and distribution, 2) education and training, 3) administration and maintenance of the database, 4) possible configurations of the database.

5.1. Implementation Issues

Assuming a decision has been made to go ahead with a database effort, implementation issues address what data goes into the database, how data will be protected, how data will be checked for correctness, what kinds of services will be available to users, and how the database will be presented. The list is not complete and only reflects what interviewees brought up during the interviews.

5.1.1. Data Definitions

Data definitions were the most frequently cited implementation issue discussed during the interviews. This was true of all sectors of the software engineering community, academia, government, or industry. Almost everyone felt that data definitions were key to setting up a successful national software engineering database; however, some suggested that the emphasis be placed on acceptable rather than exact definitions.

Data definitions provide the mechanism for establishing common terminology and the reference points for data collection and use. They also enable users and suppliers to assign unambiguous meaning to data and provide the ability to map from one data set to another. Data definitions follow from decisions about what information is to be collected and put into the database.

Participants stressed the following points:

- Data is meaningless and unusable without data definitions.
- Data definitions must be in place **before** collecting data.
- Data definitions must be accepted by everyone involved in the database effort.
- Data definitions come after decisions are made about what data to collect.

A major criticism heard directed at early data collections was that data definitions varied too much among data sources. Many potential users of data from these collections found the data to have little value; meaning, comparability, and applicability could not be

determined from the data alone. The data was regarded as suspect and unusable because questions such as these could not be answered:

- Are two data points comparable?
- How was this data measured?
- Is the data related to my situation?
- What does this number really mean?

Interviewees felt that formulating data definitions and gaining consensus for the definitions will be a major accomplishment as well as a major risk in gaining acceptance for a national database. Numerous sources related negative experiences with deriving data definitions within their own local contexts. They found the process to be time-consuming, frustrating, and difficult to accomplish successfully even within a single project, company, or program office. Many were pessimistic that it could be done nationally. Some felt that on its own, apart from all other issues, the effort of creating data definitions could forestall or terminate a national database effort. Others felt that rather than take the risk of not doing anything at all, starting with something less than perfect and gaining experience in the process offered its own advantages.

5.1.2. Confidentiality

Next to having common data definitions, interviewees thought the issue of confidentiality was of most importance to creation of a national database. Confidentiality means that names, identifiers, or any other information that could relate data back to its sources are protected. Ensuring confidentiality involves setting up policies and procedures for data security. Securing data starts from the time raw data is received from suppliers and continues through the time the data is sanitized for entry into the database, and are finally presented for access by others. Data suppliers felt that confidentiality had to be guaranteed and proven before their organizations would commit to a database effort that relied upon detailed data from their companies.

Interviewees stated their organizations would be unwilling to supply detailed data that might put them in a negative light, be subject to misinterpretation, or be misused in any way. Data suppliers wanted assurances that data for a national database would be sufficiently sanitized so that business competitors or sponsors could not extract proprietary information about products or business practices. All agreed that careful planning to protect confidentiality and maintain security is essential.

Interviewees suggested these mechanisms for protecting data:

- Remove company names, project identifiers, program names to protect the identity of the data supplier. In some cases this may not be sufficient if an application domain has few sample points or a project is unique and identifiable by its attributes.

will be screened out of the data. However, normalization presents a problem in that these very details are the ones that could determine whether data is comparable or meaningful. If not done properly, normalization could make the data useless.

- Provide only summary information in the database, e.g., averages, maximums, minimums, trends. Data can be protected at the summary level more easily than data that is more detailed. When further details are needed, use an intermediary or broker to negotiate between the source and user for any additional information that the user needs and the supplier will permit to be revealed.

5.1.3. Supporting Information

An issue closely tied to data definitions is the issue of obtaining supporting information for collected data. Interview participants felt that they needed many of the underlying details about a data point, for example, counting methods, context, application domain, experience, etc., to be able to effectively use the data. They observed that even with common data definitions, more information is needed to ensure proper use and understanding.

Interviewees illustrated the need for supporting data with examples:

- Project size: Data from small projects is different from data for large projects.

  Example: A project with 1 million SLOC is different from a project with 10,000 SLOC.

- Application domain: Data from different domains varies significantly across those domains.

  Example: Payroll systems differ from real-time embedded systems.

- Developer experience: Data from a developer with many years experience in an application domain differs from data collected from a developer with little or no experience in the same domain.

  Example: Data from a developer with 10 years experience in avionics systems is different than data from a company with one year of experience.

- Data precision: Data may be collected at varying levels of detail.

  Example: Staff hours collected to the nearest day is different from staff hours collected to the nearest month.

- Project constraints: Projects are unique especially in the constraints they are under.

  Example: DoD projects must meet standards not required of commercial projects. On the other hand, commercial projects are subject to market factors that do not affect DoD projects.

5.1.4. Validation and Verification

Validating and verifying data means asking whether the right data has been collected and whether correct collection procedures have been followed. Validation and verification are important because they ensure good quality, consistent data that may be combined with other data without corrupting the results, thereby making conclusions viable and comparisons legitimate. Experiences related by those already collecting data indicate they have a difficult time verifying and validating data. Many are forced by time, budget, or resource constraints to rely on the integrity of data suppliers to ensure the quality of the data.

Participants recommended the following techniques:

- Sampling collected data points to see if the data is reasonable within its context, that is, ballpark check.
- Training in collection methods for those responsible for on-site collection.
- Spot checking data collection sites.
- Phoning back to collection points when data looks suspect.
- Establishing audit trails whenever possible.
- Collecting redundant sample points, i.e., collecting the same data from two sources, thereby allowing comparisons to test for validity. Data reported in from both a contractor and a sponsor may be checked in this way.
- Using automated collecting tools and procedures.

Other attributes related to validation and verification that need to be considered:

- Data consistency - Consistent data is data that maintains integrity across the span of the data collection effort. Examples of consistent data are subtotals that add up to totals, and the stability of counting rules and definitions over the course of a collection effort.
- Data completeness - A complete data set is one in which all data points have been obtained to the same level of detail and there are no missing items. For example, cost data may be part of the database and data definitions may require details on environment, hardware, experience level, complexity of application, etc. An incomplete data point would be missing some of the details.

Some participants recommended that policies for administrating the database be established about handling unvalidated data. They wanted assurances that faulty data would be discarded before it could corrupt quality data One person estimated that as much as 75% of collected data might have to be rejected for one reason or another.

5.1.5. Adding Value

Interviewees were quick to point out that potential suppliers of data cannot be expected to provide data out of a sense of goodwill alone. Companies and programs need

incentives to undertake the expense, effort, and risk involved with providing data. Several data suppliers complained about previous experiences in providing data to a data collection effort, and then not getting any feedback or access to the pooled data. In the opinion of some of the interviewees, providing products and services to data suppliers helps motivate the participation of many in a database effort who might otherwise be inclined to forgo it. Besides giving value-added, this feedback and access assures suppliers that the data is being used at some level and is therefore valuable, that they are not alone in providing data, and that the data is not being misused. Providing products and services also gives visibility into the methods and procedures governing the use of the database. Some pointed out that adding value may come too late to be of much benefit in ongoing projects.

Methods for adding value include the following suggestions:

- Summary reports generated from the database. These reports could include presentation and analysis of national trends for various metrics, application domains, or programs.
- Detailed statistical analyses of the pooled data. These could include averages, benchmarks, correlations, etc., presented in textual or graphical formats.
- Recommendations to individual organizations on improvements. The organization administering the database could prepare detailed diagnostic reports for individual organizations showing where they differed from national trends and averages, and make suggestions on ways to improve.
- Training and education on how to best use the data in a local context. This could include tutorials and workshops so organizations can get the most out of using the database.
- Tools for data collection and metrics analysis. The organization administering the database could distribute tools that would help companies do their own internal data collection and analysis in standard and proven ways.

5.1.6. Access and Distribution

Access refers to who will be allowed to use the database. Some interviewees called for databases that encompass the widest scope, open to anyone desiring access, developers, sponsors, or academic researchers. Others called for more restrictive access, e.g., available only to the DoD community or only to those who supplied data.

Interviewees suggested these options for access privileges:

- Anyone who wants it.
- Only those who contribute data.
- Only those who subscribe to or pay for membership.
- By level of participation - more detailed views to those who contribute more, less detailed views to those who contribute little or none at all.

Distribution of the data refers to how data will be presented and given to users. Possibilities range from providing everything collected to providing only summaries and limited views.

Participants mentioned the following options for distribution of the data to users:

- Complete electronic copies of all data (sanitized) in a standard database format, e.g., dBase, Paradox, Ingres.
- Electronic or hard copies of data selected by predefined views of the database. Raw data would not be supplied and there would be processing and summarizing before copies of the data are distributed. Data from individual sample points would not be available. In this case those with their own data would be able to use the summaries to compare relative positions and do "sanity checks."
- Access to a central database with a predefined set of accessing and reporting operations. Copies of the database would not be distributed but open access to the database would be allowed. There could be a full or limited range of operations and views on the database depending on database policies. These views could present summary information as well as detailed information about individual sample points. All data would be sanitized.

5.2. Education and Training

Education and training were mentioned several times in the interviews. Those experienced in database efforts recommended education and training for both data suppliers and data users, involving all levels of an organization from line workers to senior management. SEL, SDDB, and the Software Data Library include varying degrees of education and training as part of their database efforts.

Benefits cited for education and training include the following:

- Getting the widest level of support from all communities involved in a database effort.
- Getting good quality data.
- Ensuring the proper use of data.
- Allaying fears and misunderstandings.

Education and training may be done in several phases and may be done to both inform and motivate. Suggested phases include the following:

- Marketing and selling the database project to senior level management or program office officials. Outline the purpose, goals, and the involvement of their organizations with the database project to those at the highest levels of authority. Inform decision makers about how both the collection and release of data will potentially impact their organizations, what mechanisms and guarantees are in place to protect their interests

and concerns, and what benefits their organization can expect in return for cooperation. Include information about costs that may be associated with the project.

- Detailed exposition of the database including database architecture, definitions, procedures, security precautions, guarantees, services etc. to those most directly responsible for actual data collection and use.
- In-depth training on collection methods, e.g., completion of report forms, use of collection tools, formats for electronic media, to those assigned to the actual collection process.
- Training for those using the data to ensure proper interpretation and usage of the data and reports generated from it. Depending upon the method of distributing information in the database, this may also include training in use of specific software and hardware for storing and retrieving information.

Several interviewees pointed out that the database itself will serve as a model for anyone involved in data collection. The architecture, data definitions, methods, services associated with the database will be viewed as examples of good practices and techniques.

5.3. Administration and Maintenance

An important issue in considering a national software engineering database is who will plan, organize, administer, and maintain the database. One interview question asked for opinions on whether the community was ready for a database ("ready" meaning mature enough to understand and make effective use of a national database). The response to this question was mainly "yes" but with many qualifications about how the data is likely to be used or misused. A follow-up question asked for a direct response about the role of the SEI in relation to a national database. The question asked "Should a national database be something the SEI should do?" Interviewees gave a positive endorsement to the SEI having some role in future discussions about a national database; however, it should be emphasized the SEI has no plans associated with a national database.

Interviewees pointed out that risks for the task are great and failure of a database effort may seriously damage the reputation of those associated with it. Negative experiences with earlier data collection efforts have persisted long after changes and corrections have been made. A new effort may fail in several ways: 1) the effort falters because of poor planning, insufficient resource allocation, or inadequate funding, 2) no one uses the data because it is irrelevant, ambiguous, corrupt, or out of date, 3) security is breached, 4) data is misused.

5.4. Database Configurations

Database configurations explores various possibilities about who participates in the database effort and what roles the participants take. Interviewees had several proposals to make about organizing a database. The proposals include government/industry, DoD/DoD-related industry, government/industry/academia, all industry, specific application domains, or functional domains.

Government/Industry

In this version of a database, the federal government cooperates with the software industry to set up and finance a database. The academic community is excluded from participation. Government participants include the entire DoD, other federal agencies like NASA, the Federal Aviation Administration (FAA), or Government Accounting Office (GAO). Industry participation includes both commercial and DoD-related industries.

DoD/DoD-related industry

This is a restricted version of the previous organization in which only the DoD and DoD-related industries participate. Since most DoD contracts follow required standards and procedures, coordinating a database effort might be easier than trying to include all industry. On the other hand, a DoD-sponsored database might be seen as a way for the DoD to obtain information to control contractors, and it may not be welcomed by contractors.

Industry/Academia

A database organized in this fashion requires a cooperative effort between industry and academia, leaving the government out altogether. Variations on this approach include setting up regional databases administered and maintained at academic centers with data provided by local developers. The individual who proposed this type of organization suggested that issues of confidentiality and trust might be alleviated by the nearness and familiarity of database administrators. In other words, it is easier to work with and trust someone you know and can talk to than with a central, faceless, unfamiliar group. SERC and SEL are examples of how this cooperation can work. Academia provides the personnel, resources, expertise in experimental methods and data analysis to the effort while industry provides the raw data and experience in software development.

Government/Industry/Academia

In this arrangement all three communities combine to bring the best of their expertise to the effort. Industry and government provide experience, data and financial resources. Academia brings research expertise and staff for analyzing data, running experiments, and publishing results.

Industry Only

In this type of organization the software industry, both DoD and commercial sectors, takes charge of planning, financing and running the database. The software industry is then able to set up its own standards, guidelines and benchmarks. Peer pressure plays a major role in ensuring support and participation.

Application Domains

A database organized around application domains is one in which a single domain such as the telecommunications or the banking industry participates. This has already been done by the aerospace industry with the SDDB database. The advantages to this way of organizing is that it limits the scope of the database and makes it more tractable than a database including everyone. Some of the advantages are a common purpose and sharing of common problems. A major disadvantages is that it gives a narrow focus, and is affected by direct competition among the member companies.

Functional Domains

Databases set up by functional domains are organized around a specific metric or type of data, for example, reliability, cost, and quality data. SDDB is basically a cost database and an example of this type of database. The advantage to organizing by functional domains is that data for a specific purpose is collected and used. The disadvantage is that other data is not incorporated or looked at in combination, e.g., there may be interactions between reliability and quality that are missed by isolating the databases.

Combinations

Several persons interviewed suggested that smaller databases of limited scope, similar to any of the ones mentioned above, could be combined into supernetworks of databases with all the advantages and disadvantages of distributed or hierarchical systems.

6. Concerns

While answering questions during the interviews, participants expressed a variety of concerns or reservations about a national database. Concerns reflect broader, more philosophical aspects of the database as contrasted with the more specific, pragmatic issues Chapter 5 describes. Chapter 6 presents participants' concerns about data uses, data sources, database costs, data collection mandates, and use of data for evaluation purposes.

6.1. Uses

Most practitioners and researchers interviewed felt that the software engineering community was ready to accept the idea of a national database; however, many had reservations about whether the software community was mature enough to use the database effectively. They made the following points about data usage:

- Companies are unsuccessful in organizing, collecting, and understanding their own data. It is naive to think national data will make up for what is missing internally.
- Looking at a company's data is like looking at their accounting records; business practices differ between companies and the numbers mean different things. Companies have their own reasons for being in business and this affects how they do their accounting. It also affects how and why they collect software engineering data.
- Meaningful interpretations of data assume commonality of purpose, definitions, counting rules, collection methods, etc. The current state of the practice does not justify these assumptions; there are no standards, no common definitions, no proven data collecting techniques in software engineering practice.
- Collecting data is relatively easy when compared to using and interpreting the data. Most organizations are not ready to understand and use national data.

Several interviewees strongly recommended encouraging organizations to start their own internal data collections as a more sensible alternative to relying upon a national database for data needs. They suggested that money would be better spent training organizations to do their own data collection. Interviewees observed that as more organizations begin the practice of data collection then common pooling of data is likely to happen spontaneously, especially when organizations begin to feel the need to communicate with one another over results and findings.

6.2. Data Sources

Everyone had concerns about the quantity and quality of data supplied to the database. This concern was directed at both the kinds of companies supplying data and the numbers of sources for data. There are many more potential users of a database than there are data

providers (there was one guess at 90% users, 10% suppliers). The following concerns were expressed over obtaining data to populate the database:

- Relatively speaking, not very many companies are collecting data, and data collection is not a high priority item yet with management. There is a cost to collecting data and in many instances the data has questionable value to managers. Companies often stop data collection when schedules slip or budgets are over-spent because they have not seen sufficient pay back for the collection effort.
- Even fewer companies are willing to share data. Sharing internal company data is not an accepted part of software engineering practice. Companies are reluctant to give competitors any information that will weaken a competitive position or make them look incompetent.
- Few companies have good data. There are estimates that 85% of the developer organizations have unrepeatable software development processes. These processes will likely generate poor quality data that could corrupt valid data in the database. Interviewees noted that mixing bad data with good data affects both the credibility of the database and any data analysis.
- Data is not compatible across companies. Every company has its own objectives for doing business and reasons for collecting data. Thus, the resulting data items vary too much among companies to be of any value for common data collections. For example, some companies develop software to sell to a market; others develop software to fulfill a contract.
- Some domains have too few sample points from which to draw data to make the data collection worthwhile. For example, in the domain of operating systems development there are less than 10 major developers.
- Company profiles vary too much even within similar domains. Profile characteristics including the size of a company, years of experience in the domain, and strategic goals of the companies can affect data from the same domain. For example, data from a new start-up company is not the same as data from a company that has been in the business for years.
- Data ages too quickly. With rapid changes in software engineering technology, data collected over lengthy projects may not be relevant in the long term. For example, data from a 5-year project is likely to be out of date by completion of the project. Technology infusion during a project may also change the meaning of data collected during the project. For example, introducing a formal review process mid-stream in a project will affect quality data.

Increasing the pressure to improve development processes and initiate measurement programs would put many organizations in better positions to collect, supply and use data in the future. Some participants see this as a necessary first step to guaranteeing a sufficient supply of data for a national database.

6.3. Data Collection Mandates

Several representatives of both government and contractor organizations discussed data collection mandates during the interviews. Members of both groups had reservations about mandates and thought they should be avoided as a way to populate a database.

Data collection is mandated when a sponsor or customer requires a contractor to collect data and report out the data as part of a software contract agreement. Sponsors and contractors did not consider mandates a desirable practice since mandates can be counter-productive, leading to fabricated data, wasted money and effort. With mandates there is a question over who pays for the data collection and reporting -- the sponsor or the contractor. Few thought sponsors would be able to fund data collection considering current government fiscal reductions and would expect contractors to absorb the costs internally. Contractors were concerned over how to cover the costs of collecting data under their own tight budgets.

6.4. Cost and Resources

Interviewees had concerns over financial support and allocation of resources for a national database. These concerns were directed particularly at the costs and resources needed by the host or administering organization to collect data, enter it into the database, and provide database services. There was a general feeling that a national database would be too expensive and there would not be sufficient resources, e.g., hardware, personnel committed to it.

Several people involved with ongoing database efforts indicated that in actuality the costs and resources for organizing and maintaining a national database might not be as high as anticipated. Data collection costs can be kept down by using data already being gathered by organizations and by providing automated collection tools. Hardware requirements, particularly for transfer and storage media, may not be as large or expensive as expected either. Examples were given of putting an entire database on a diskette or magnetic tape. One database effort was able to keep staff to one or two people, and costs well below $1 million for the entire project over several years.

There was also concern about the stability and longevity of the database effort. This included maintaining funding at adequate levels and providing long term commitments of resources. Several database efforts have already faced this problem; some went out of existence, others, tried to do their best under the circumstances.

6.5. Using Data for Evaluation Purposes

Developers and contractors were concerned about possible use of data for rating or evaluating their organizations. This was seen as something that would most likely happen in awarding contracts, that is, sponsors would use the data for picking the "best" contractor.

There does not appear to be any fool-proof resolution of this concern except to recognize that it exists and to take steps to prevent it from happening. For example, a database with only summary data could alleviate this situation. One individual warned that using the data to evaluate companies would be a certain way to lose voluntary participation.

7. Advice

The interview process solicited advice to take advantage of participants' experience and expertise in using and collecting data. Comments made during the interviews are paraphrased below. The text is prescriptive in tone and reflects the way advice was given in the interviews. Some of the advice is conflicting and no attempt has been made to resolve these conflicts. Readers are encouraged to make their own evaluations.

7.1. The Basics

Everyone agreed that getting a national database effort started is a complex and difficult task. The most commonly heard words of advice were:

KEEP IT SIMPLE.

Variations on this theme:

Start small.

Prototype first.

Behind this advice is the notion that simple, small efforts are easier to sell, plan for, and administrate. Trying out a pilot or prototype database makes it easier for participants to say "yes" to since initial costs and involvement are relatively low. A small initial effort gives people a chance to try out the database, see what it can do, and see what advantages it offers, without investing a lot of time and resources. The success of a modest effort is likely to "grow" support and enthusiasm for further refinements.

Interviewees warned against starting out with large databases that try to do everything. As one person put it, "It's like trying to swallow an elephant whole", and described a complex project as a sure way to kill a database effort. The cancellation of the Software Data Library was cited as an example of what can happen when too much is attempted at once.

7.2. Planning

Data collectors, suppliers, and users all urged that a database effort be well-planned in advance of any actual data collection. Planning includes all aspects of setting up a database, i.e., data definitions, database model, formats, access methods and privileges, data sources, data analysis and report generation. Planning also involves setting up clear-cut goals and objectives for the database. Funding, cost estimations, staff requirements should also be taken into account when planning.

Planning advice:

- Recognize this is a complex task and do careful planning in advance. Establish clear-cut goals and objectives before deciding what data to collect.
- Plan for adequate staff and resources, and provide sufficient funding to carry the effort through. Bite the bullet on resources to get the best, most technically competent people needed. This is not an easy task and should not be done cheaply.
- Plan to throw the first database away. No matter how much planning is done, something is always left out.
- Be flexible in planning. There are too many factors that can change even in a short time, for example, needs, definitions, or infusion of new technology.
- Plan to allocate enough funds for verification and validation. This is an aspect omitted in funding for many existing databases.
- Use an existing database as a starting point and go on from there.
- Try planning varying levels of databases, that is, high, medium, low levels of detail. This will aid in future planning providing a framework for a larger database.
- Have plans reviewed by as many people as possible to get feedback and enlist more widespread support.

7.3. Data Definitions

Good, clear, universally acceptable data definitions were one of the most frequently expressed requirements stated during the interviews. Chapter 5 covers data definitions in detail, but they are mentioned again here because of the repeated admonitions from interviewees to get them right. Advice on data definitions includes the following:

- Schedule forums at conferences, workshops, etc., to discuss and formulate definitions.
- Create a working group to formulate definitions. This is something the organization supporting the database effort can do.
- Use existing standards and definitions, for example, IEEE and ISO. Do not waste time and effort doing separate definitions.
- Provide automated collecting tools. This will ensure conformance of the data to the definitions.
- Provide templates and software packages to facilitate manual data entry.
- Make the definitions goal-oriented, that is, use the GQM model to set up definitions. This will help keep definitions focused.
- Be selective about data; do not take just anything. **Bad data will destroy the credibility of the database.**

7.4. Gaining Support

A database needs data to be successful. Populating a database with samples will be a problem if organizations that generate data do not support the database effort. Advice on how to gain support for a national database includes the following:

- Get support from top management. Frequently cited examples of lessons learned in data collection stressed the importance of getting top level support BEFORE launching a database effort.
- Institute participation as a standard part of doing business. Look for models in other industries as examples, for example, Association for Industrial Purchasing Agents, the American Refrigeration Industry.
- Use peer pressure to gain support. Organizations will want to participate if they see competitors involved and gaining some advantage, that is, publicity, recognition, technical support, or privileged information.
- Make sure there is a pay back. Companies are more likely to want to participate if there is something in it for them in return. Technical expertise, data analysis, and individual reports are examples of value-added features.

7.5. Trust and Confidentiality

Confidentiality was a major issue with data suppliers. Advice offered about methods and procedures for ensuring confidentiality includes the following:

- Trust is something that has to be gained. Promising to protect confidentiality does not carry the same weight as actually doing it.
- Start with a small effort and demonstrate trustworthiness, i.e., collect some confidential data, use it, and show that those in charge can be trusted.
- Keep confidential data in the hands of as few people as possible.
- Carefully screen people handling sensitive data.
- Outline in advance mechanisms for protecting confidentiality.
- Breaches in confidentiality cannot be repaired.

8. Conclusions

The following conclusions were drawn from the study:

- A national database will be much easier to implement when measurement practices become standardized across the industry.
- A national database with industry-wide macro-level information is achievable if someone wants to make the effort to locate and consolidate information available in the public record.
- A national database at a detailed, micro level is not feasible at this time.
- A database supporting new technologies and methodologies is not feasible at this time.
- An artifacts database has little support from industry at this time and needs more careful study to justify the expenses and risks involved.

A common theme emerging during the interviews was the requirement for data definitions to ensure the success of a national database. Data definitions came up as a subject in almost every interview as either an issue, a concern or the target of advice. Reservations were also expressed about the maturity of the software community to populate and effectively use a database given the current state of software engineering practice. Both of these factors point to larger community issues of putting appropriate foundations of standardized software measurement practices in place before starting a national database.

An even larger issue to the software community is that of putting data and databases into proper perspective as means to an end and not ends in themselves. The software community needs to understand the role of data as a tool to understanding the software process and sustaining continuous process improvement. It is all too easy to become involved with the details and loose sight of this important fact.

The database picture is not all that bleak, and interviewees pointed out that there were realizable goals for national data that were relatively easy to achieve. In particular a macro-level database containing industry-wide data already in the public record could be assembled rather easily. Such a database would be useful to the industry as a whole for getting a clearer picture of the national software industry and for bringing the idea of a database into practice and an accepted part of software development culture.

A database with more detailed information will require more time and further maturing of the software industry. Both the micro-level database and the database supporting new technologies and methodologies are included in this category since both require data at similar levels of detail. Measurement programs, measurement definitions, data collection methods need to be clearly defined and outlined before such an effort can be feasible. Interview participants pointed out that where data collection and measurement are in practice, measurement methods and data definitions vary widely between organizations and even within organizations. Populating and using a national database requires that data be

collected consistently and understood unambiguously to avoid problems of misuse and misinterpretation. Advice from interviewees reinforced these concerns and recommended initial database efforts be simple, well-defined, and clearly focused. This advice was supported by the evidence of successful software engineering databases whose success has been achieved by limited scopes and modest goals.

An artifacts database is something many in the industry felt was unachievable and impractical in the near term. A major obstacle is the reluctance of industry to share code and proprietary data on a public basis. The special needs of researchers for artifact data are being addressed in other venues. The National Science Foundation (NSF) in January 1990 sponsored a workshop that established a channel of communication between industry and academic researchers to understand each other's needs and seek methods of accommodation.

There are many working relationships within industry and between industry and academia that have been successful. The Software Engineering Laboratory (SEL) in Maryland, with NASA/Goddard, Computer Sciences Corporation and the University of Maryland, and the Software Engineering Research Council (SERC) at Purdue University and a group of 16 local industries are two conspicuous examples of how things can be worked out successfully. There are also many other examples of industry cooperating with individual researchers, e.g., Bellcore and AT&T Labs.

On an optimistic last note, as measurement programs take hold in industry and data definitions and data collection methods become standardized, many of the obstacles to a national database will disappear. Practitioners collecting data will be eager to discuss results and lessons learned with others and to share their data. Then common data pooling and larger, national databases will become more tractable and eventually be accepted as part of the culture of the software community.

References

[Basili 84] Basili, Victor R.; & Weiss, David M. "A Methodology for Collecting Valid Software Engineering Data." *IEEE Transactions on Software Engineering 10*, 6 (November 1984): 728-738.

[Beaver 91] Beaver, Janet K.; O'Neill, Patrick J.; & Betz, Henry P. "U.S. Army Software Test and Evaluation Panel (STEP) Software Metrics." *Proceedings of the Annual Oregon Workshop on Software Metrics*, Silver Falls, Oregon: 1991.

[Boyer 88] Boyer, M.; & Simmonds, I. *Project Final Report* (Software Data Library Report). United Kingdom: National Computing Centre Limited, 1988.

[DACS 91] *DACSGUIDE: A User's Guide to DACS Products and Services.* Utica, N.Y.: Data and Analysis Center for Software, 1991.

[Date 90] Date, C.J. *Database Design.* New York, N.Y.: Addison-Wesley, 1990.

[DoD 91] *Department of Defense Software Technology Strategy* (draft). December 1991.

[IEEE 90] *Standard for Productivity Metrics* [draft], (P1045/D4.0). Washington, D.C.: Institute of Electrical and Electronic Engineers, Inc., December 20, 1990.

[Grady 87] Grady, R.B.; & Caswell, D.L. *Software Metrics: Establishing a Company-Wide Program.* Englewood Cliffs, NJ: Prentice-Hall, 1987.

[Knuth 71] Knuth, Donald A. "An Empirical Study of FORTRAN Programs." (RC3276) Yorktown Heights, N.Y.: IBM Research, March 8, 1971.

[RADC 88] Rome Air Development Center. *Automated Measurement System* (RADC Technical Report TR-88-101). Rome, N.Y.: Roma Air Development Center, 1988.

[SEL 89] National Aeronautics and Space Administration. *Annotated Bibliography of Software Engineering Laboratory Literature* (SEL-82-806). Greenbelt, Md: Software Engineering Laboratory, November 1989.

[SWDL 87] *Initial Contact Presentation* (Software Data Library Report). United Kingdom: National Computing Centre Limited, 1987.

[Valett 89] Valett, John D.; & McGarry, Frank E. "A Summary of Software Measurement Experiences in the Software Engineering Laboratory." *The Journal of Systems and Software 9*, 2 (February 1989):137-148.

[Yu 88] Yu, T. J.; Nejmeh, B. A.; Dunsmore, H. E.; & Shen, V. Y.."SMDC: An Interactive Software Metrics Data Collection and Analysis System." *The Journal of Systems and Software 8*,1 (January 1988): 39-46.

Appendix A: Glossary

Acronyms

COCOMO	Constructive Cost Model
CSC	computer software component
CSCI	computer software configuration item
DACS	Data Analysis Center for Software
DARPA	Defense Advanced Research Projects Agency
DoD	Department of Defense
ESC	Electronic Systems Center
GSFC	Goddard Space Flight Center
IEEE	The Institute of Electrical and Electronics Engineers, Inc.
LOC	lines of code
NASA	National Aeronautics and Space Administration
SSCAG	Space Systems Cost Analysis Group
SEI	Software Engineering Institute
SLOC	source lines of code
SMC	Space and Missile Systems Center

Terms Used

Terms used in this document are explained below: the explanations are the author's.

Data collection - a collection of objects that may or may not be organized by any structured computerized methods. A collection denotes a set of data accumulated in some form, i.e., paper, electronic media, or a combination of both. For example, the data assembled by payroll, cost analysts, and first level managers in a development organization would comprise a data collection.

Data repository - a storage place for similar data objects, e.g., a repository of reusable code, a repository of Ada packages, a repository of current best practices.

Database - a computerized system whose overall purpose is to maintain information and to make that information available on demand [Date 90]. This definition, used in discussing databases and database systems, includes the concepts of data models, query languages, data management, and database architectures. This report will use a less formal definition to talk about a software engineering database, and will consider a software engineering database to be loosely a set of persistent (enduring over time) software engineering data elements.

National - national by definition means belonging to or relating to a nation as a whole. A national software engineering database contains data that is relevant to the software engineering industry of the nation.

Software engineering data - information pertaining to the practice of software engineering, e.g. artifacts, measurements, practices, processes, and products.

Appendix B: Interview Forms

National Database Study

Background Data

Name: _____

Company: _____

Title: _____

Date: _____

Phone Number: _____

Company's Business:

A.	Developer/Contractor	[ ]
B.	Acquisition	[ ]
C.	Research	[ ]

Community:

A.	Government	[ ]
B.	Industry	[ ]
C.	Academia	[ ]

Data Needs:

A.	User	[ ]
B.	Supplier	[ ]
C.	Collector	[ ]

General interview questions regarding a national database:

Do you see a need for a national database (a specific set of defined data items)? Generally or for a specific need of yours?

What should be the goal of a national database?

What are the questions relevant to that goal?

What data should be collected?

Are we (SE community) ready for a national database?

Should this be something that the SEI should do?

Are there any alternatives?

From your experience in dealing with data, what advice in general do you have?

Suppliers

Is a national database of any value to your company? Why?

As a potential supplier of data for a national database, what concerns do you have specific to your company, your role in that company?

Would management be willing to release data to a national database?

What kind of data? (general, specific, low level)

Would having data definitions affect your willingness to contribute?

Collectors

What is the history of your data collection effort?

What is the purpose of your database effort?

What questions are being asked?

What data is being collected?

How is it being collected?

Are data definitions being used?

How big is your collection? ("Number of records")

Who contributes data?

How do you ensure validity of the data?

What services do you provide users/suppliers?

How many users are there?

How is the data used?

Are there users who are not suppliers?

How do you screen them?

Has your effort been successful? Why?

What lessons have you learned?

What advice do you have?

What concerns do you have?

Any publications describing your effort, lessons learned, history, etc.?

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION Unclassified	1b. RESTRICTIVE MARKINGS None
2a. SECURITY CLASSIFICATION AUTHORITY N/A	3. DISTRIBUTION/AVAILABILITY OF REPORT Approved for Public Release Distribution Unlimited
2b. DECLASSIFICATION/DOWNGRADING SCHEDULE N/A	
4. PERFORMING ORGANIZATION REPORT NUMBER(S CMU/SEI-92-TR-23	5. MONITORING ORGANIZATION REPORT NUMBER(S) ESD-TR-92-023

6a. NAME OF PERFORMING ORGANIZATION Software Engineering Institute	6b. OFFICE SYMBOL (if applicable) SEI	7a. NAME OF MONITORING ORGANIZATION SEI Joint Program Office
6c. ADDRESS (City, State and ZIP Code) Carnegie Mellon University Pittsburgh PA 15213		7b. ADDRESS (City, State and ZIP Code) ESD/AVS Hanscom Air Force Base, MA 01731
8a. NAME OFFUNDING/SPONSORING ORGANIZATION SEI Joint Program Office	8b. OFFICE SYMBOL (if applicable) ESD/AVS	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER F1962890C0003
8c. ADDRESS (City, State and ZIP Code) Carnegie Mellon University Pittsburgh PA 15213		10. SOURCE OF FUNDING NOS. PROGRAM ELEMENT NO: 63756E; PROJECT NO.: N/A; TASK NO: N/A; WORK UNIT NO.: N/A

11. TITLE (Include Security Classification)
A Concept Study for a National Software Engineering Database

12. PERSONAL AUTHOR(S)
Patricia B. Van Verth

13a. TYPE OF REPORT Final	13b. TIME COVERED FROM TO	14. DATE OF REPORT (Yr., Mo., Day) July 1992	15. PAGE COUNT 61

16. SUPPLEMENTARY NOTATION

17. COSATI CODES			18. SUBJECT TERMS (Continue on reverse of necessary and identify by block number)
FIELD	GROUP	SUB. GR.	
			national software engineering database software engineering database software metrics software measurement

19. ABSTRACT (Continue on reverse if necessary and identify by block number)

Substantial segments of the software engineering community perceive a need for high-quality software engineering data at a national level. A national software engineering database has been proposed as one way to satisfy this need. But is such a database feasible and can it really satisfy these needs?

This report provides information obtained from an informal survey of members of the software engineering community about a national database. The survey served as a means of getting a sense of the expectations that are to be met by experiences of those surveyed about data collection and use. The report summarizes this material in a manner that is informative and expository rather than prescriptive.

(please turn over)

20. DISTRIBUTION/AVAILABILITY OF ABSTRACT UNCLASSIFIED/UNLIMITED SAME AS RPTDTIC USERS ■	21. ABSTRACT SECURITY CLASSIFICATION Unclassified, Unlimited Distribution

22a. NAME OF RESPONSIBLE INDIVIDUAL John S. Herman, Capt, USAF	22b. TELEPHONE NUMBER (Include Area Code) (412) 268-7631	22c. OFFICE SYMBOL ESD/AVS (SEI)