Estimating Data Reverse Engineering Projects

by

Peter Aiken✳✤ and Pam Piper✤

✳ Department of Information Systems
School of Business
Virginia Commonwealth University
Richmond, VA 23284-4000 USA

✤ Defense Information Systems Agency
Center for Information Management
Data Administration Program Management Office
1951 Kidwell Drive, Vienna, VA 22182 USA

Overview

Systems reengineering projects are defined by the process of first reverse engineering appropriate components of the target system and then forward engineering it using the knowledge gained from the reverse engineering efforts [Chikofsky & Cross 1990]. The cost of the reverse engineering portion(s) can be a major determinate of the ultimate viability of the reengineering effort. This paper describes an approach to developing reverse engineering project estimates based on our experience with several Department of Defense data reverse engineering efforts. Like all analysis problems, useful project estimates can be best established only after preliminary study of the project context. As a result, it is necessary to invest a relatively small amount of resources in a preliminary system survey prior to attempting to develop project estimates. The information gained from this approach can be used to better evaluate the overall system reengineering costs.

Outline

1.6 References

1.1 Introduction

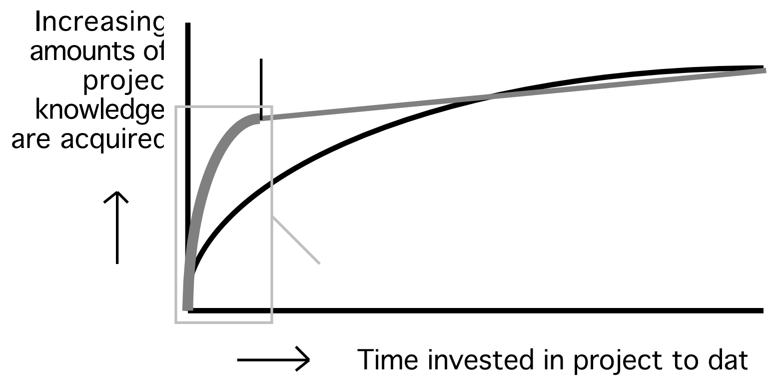

Figure 1 The preliminary system survey attempts to acquire project information more rapidly than the traditional means of project information acquisition.

The adjective *useful* is often paired with the noun 'estimate' implying some estimates are useful and some are not. Analysis problems pose dilemmas for organizations because organizations like to know how much proposed (re)development will cost *at the time when the least is known about the nature of the project!* The solid line in Figure 1 illustrates this more traditional approach to estimating data reverse engineering projects of first 'guestimating' the project and then beginning work on it. Estimates made from insufficient system knowledge are not often useful. As soon as work begins, the project team begins to amass considerable amounts of information about the project. In the case of data reverse engineering analyses, the solution is to perform a preliminary system survey (PSS) of the proposed system to rapidly gather information about four key project variables characterizing individual reverse engineering projects. (The PSS is shown as the wider gray line in Figure 1.) This information must then be combined with organizational reverse engineering productivity data and other situation specific considerations to develop project estimates and determine project feasibility. This approach to project estimation is illustrated in Figure 2. The next section of this paper describes the four major data reverse engineering project characteristics as comprised of specific facts. Section three describes the implementation of the preliminary system survey. This is followed by a description of process of combining these project specific characteristics with organizational productivity information into a project estimate. The paper concludes with some additional considerations and plans for future research.

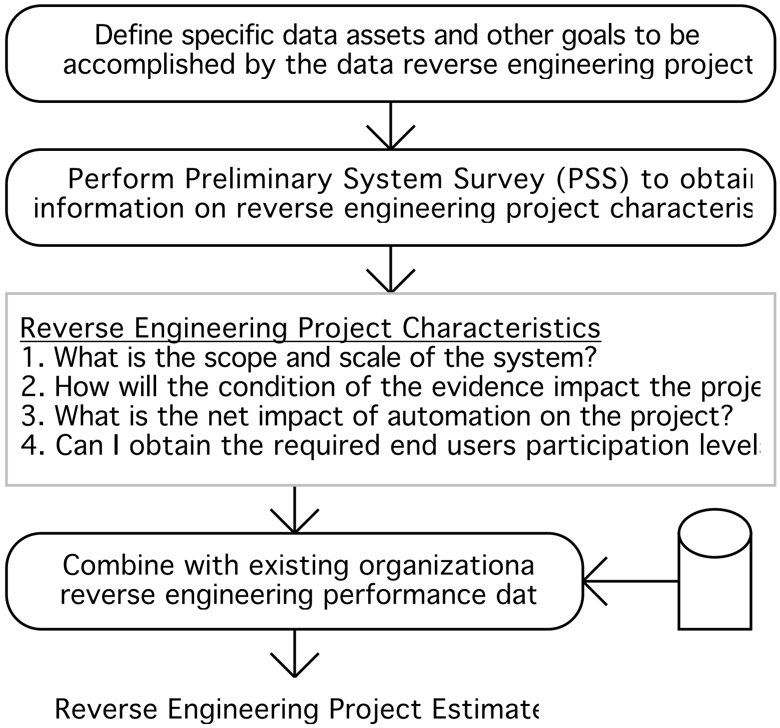

Figure 2 The process of obtaining useful reverse engineering project estimates.

1.2 Data Reverse Engineering Project Characteristics

As shown in Figure 3, four project specific characteristics are derived and assessed to determine project characteristics. Variations in:

1. system size
2. the condition of the evidence
3. participation levels of key personnel, and
4. the net impact of automation

combine in various ways to create individual project characteristics. Understanding these characteristics is key to developing useful project estimates. While no component has total dominance over the others influence can be strong. For example, it is possible for the participation component to render a proposed project infeasible. (This situation has occurred to the authors when key end users have been unavailable to participate on a data reverse engineering project because they were more urgently needed on a forward development effort occurring simultaneously.) The four project characteristics are described in more detail in the sections below.

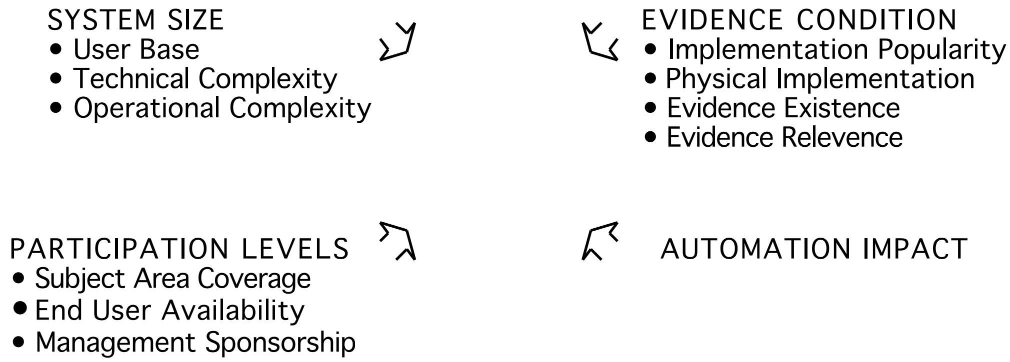

Figure 3 Four Data Reverse Engineering Project Characteristics Describe Data Reverse Engineering Projects

1.2.1 System Size

Component	**Desired Result**	**Characteristic Range or Dimension**
System User Base	An understanding of both the number and type of system users.	The diversity of the system users indicates the size of the task of comprehending the diversity. Large user populations imply less problems obtaining end user participation.
System Technical Complexity	An understanding of the system's technological implementation details including: the complexity of the system algorithms; the programming language(s) or operating system(s), and data structures.	Complex technological implementations imply more difficult analysis. The system may be technically complex as typified by the airline reservation system or it may be an unsophisticated batch-transaction processing program. Practitioners may find McCabe's work in this area useful [1976].
System Operational Complexity	An understanding of the system's host environment and implementation details.	System features such as: lack of a macro language; 24-hour daily availability requirements, or the fact that system communication must occur via 8-track magnetic tapes can force additional requirements on any subsequent data reverse engineering analysis.

Figure 4 Project Characteristic: *System Size*

This characteristic provides gross measures of system scope and complexity by determining the relevant dimensions of the system to be reengineered. These dimensions enable readers to understand the dimensions of system physical implementation, user and technology base, and system procedural operational and technical complexity. Important dimensions include:

- Is it a small compact system used by a small group of users (such as an group decision support system)?
- Is it a small compact system used by thousands of users (such as an automatic teller machine control program)?

- Is it a large distributed inter-organizational information system such as an airline reservation system?
- Does the system produce paychecks from a transaction type input or does the it handle all of the accounting functions related to the production of payroll, including IRS reporting.

1.2.2 Evidence Condition

Component	**Desired Result**	**Characteristic Range or Dimension**
System Language/ Operating System Popularity	An understanding of the degree to which both the degree of obscurity and availability of knowledgeable technical support staff will impact the project.	Less 'popular' technological implementations imply a smaller degree of tool and technical support. For example, there are lots of COBOL based reverse engineering products and considerably fewer for EDISON.
Physical Implementation	An understanding of the volume of evidence requiring analysis - typically includes such items as pages of code, file based data items, reports, screens, fields, data structures, schemas, business rules.	The physical implementation can help or hinder in various ways but existence is preferable to non-existence. The existence of 500,000 lines of code in hard copy format might be considered a barrier to effective code analysis. The same 500,000 lines of code stored in ASCII on a CD ROM would be considered more useful evidence.
Evidence Existence	A checklist of all evidence categorized as existent available, and verified.	Existence of evidence is naturally good news to the data reverse engineering analysts.
Evidence Relevance	An assessment of the state of the system documentation as: • in synchronization with the current state of the system • not in synchronization with the system state	Working with but unawareness of unsynchronized evidence can lead to much confusion. Evidence that is synchronized with the current system state is better but it still may not be able to provide the desired information.

Figure 5 Project Characteristic: *Evidence Condition*

This characteristic factors is a measure of the expected contribution of the evidence to the overall project information and determines how it impact the project. These components describe the task associated with performing the analysis of the existing evidence. System documentation ranges from not useful to useful. Where in between the two extremes does this system lie? Are they completely undocumented? Or has the organization been able to apply the resources required to keep the documentation synchronized with the system changes? Some evidence exists only in the heads of end users. Additional factors such as the state of organizational knowledge the system technology, version control and configuration management procedures must also be factored in. Each piece of desired evidence as categorized as available or not available. The assessment is best accomplished by examining the documentation first hand, performing a series of comparisons with the existing system to verify correctness, and following the change history with the documentation changes to get a feel for the system volatility.

1.2.3 Participation Levels

Component	**Desired Result**	**Characteristic Range or Dimension**
Subject Area Coverage	An understanding of the comprehensiveness of user understanding of relevant aspects of the system and its implementation.	User understanding of systems can range from no understanding through sophisticated understanding (such as awareness of certain database insertion and deletion anomalies).
End User Availability	An understanding of the likely availability of key end users required to achieve project objectives.	Useful user participation ranges from full project team member to a guest subject area expert. Project estimates are based on and tied to the specified levels of participation of specific end users.
Management Sponsorship	An understanding of the nature and status of management support for the project.	Useful management participation can range from sponsorship to champion. Perhaps the least desirable circumstances occurred when management didn't understand the project context.

Figure 6 Project Characteristic: *Participation Level*

Often participation of individuals can be used to leverage the data reverse engineering analysis. As shown in Figure 6, the user participation characteristic is comprised of: subject area coverage, key end user availability, and management sponsorship. This project characteristic describes the specific participation levels required of specific individuals who possess valuable system knowledge and will play valuable roles in the reverse engineering analysis (see Figure 7). It addressees issues such as:

- the desirability of transporting (virtually or physically) end users who work in different part of the world - are they available?
- Multiple stakeholders can create conditions where it is important to consider all perspectives - have they all been identified?
- Appropriate levels of participation by key end users including requisite management sponsorship. Note: appropriate means enough participation to ensure wise investment is made of organizational resources.

As mentioned above, obtaining adequate key user participation levels can be crucial to project success. Levels of end user participation above the project cost justification point can increase analysis effectiveness and efficiency up to a point of diminishing returns brought about by communications difficulties.

(INSERT SAMPLE FORM DESCRIBING REQUIRED USER PARTICIPATION)

Figure 7 Sample Representation of User Participation Requirements

1.2.4 Automation Impact

The last primary factor attempts to assets automation impacts available and directly applicable to the goals of the project. To what degree will automation in the form of CASE technology or other tools contribute to the effectiveness and efficiency of the analyses? The answer will range from:

- No Support: no automated tools support the platform/language combination
- Fully Supported: the popularity of the system created a market for the automated tool support that can be used to accomplish certain specific tasks.

1.3 Developing Preliminary System Survey Information

It is more useful to answer the question: "How much will is cost me to data reverse engineer system X?” with the response:

> "It will cost you at least the cost of a preliminary system survey to find out! useful project estimates can be developed after the system survey is complete.

than it is to make uninformed judgments that might return to haunt the project sometime in the future. By definition, data reverse engineering projects are explorations - Chikofsky has described them as archaeological in nature [1990]. This analogy extends to the preliminary system survey (PSS) - an initial inquiry as to the nature of the system to be examined by the reverse engineering analysis. The PSS goal is to develop an accurate baseline assessment of the project characteristics required to obtain the desired data assets [Aiken 1995]. Useful data reverse engineering project estimates can be developed if adequate PSS information is obtained about the project: participation, size, evidence, and automation characteristics. Figure 8 illustrates the steps/objectives to be accomplished when developing PSS information.

1. Determine and catalog specific information required to accurately estimate the time, personnel and other resource requirements for the project.
2. Assess and catalog the quality and availability of the existing evidence.
3. Synthesize the available information into meaningful project size, evidence, participation and automation characteristics.
4. Where possible interpolate, derive, infer or otherwise obtain and/or correct missing or substandard information.
5. Document the process - especially actions taken in step #4.

Figure 8 Sequential Preliminary System Survey Steps/Objectives

A system survey can last from several hours to 30 days and several person months of effort. An overview of the process of accomplishing a medium sized PSS is shown in Figure 9. The survey begins by identifying and securing the participation of end users. Failure to acquire their participation at the minimum required levels is a primary cause of data reverse engineering project failure. Executive management is an end user of all organizational systems because they are directly affected by the degree of system support of organizational objectives. Thus for executive management the participation is defined as active sponsorship. Having secured end user participation the performance team focuses on developing and validating a systems context model and functional decomposition. This is accomplished through a series of work group session where data assets are refined and validated. The four project characteristics become the first organizational data assets created from the candidate data reverse engineering project and should be stored in the project data bank for anticipated future use.

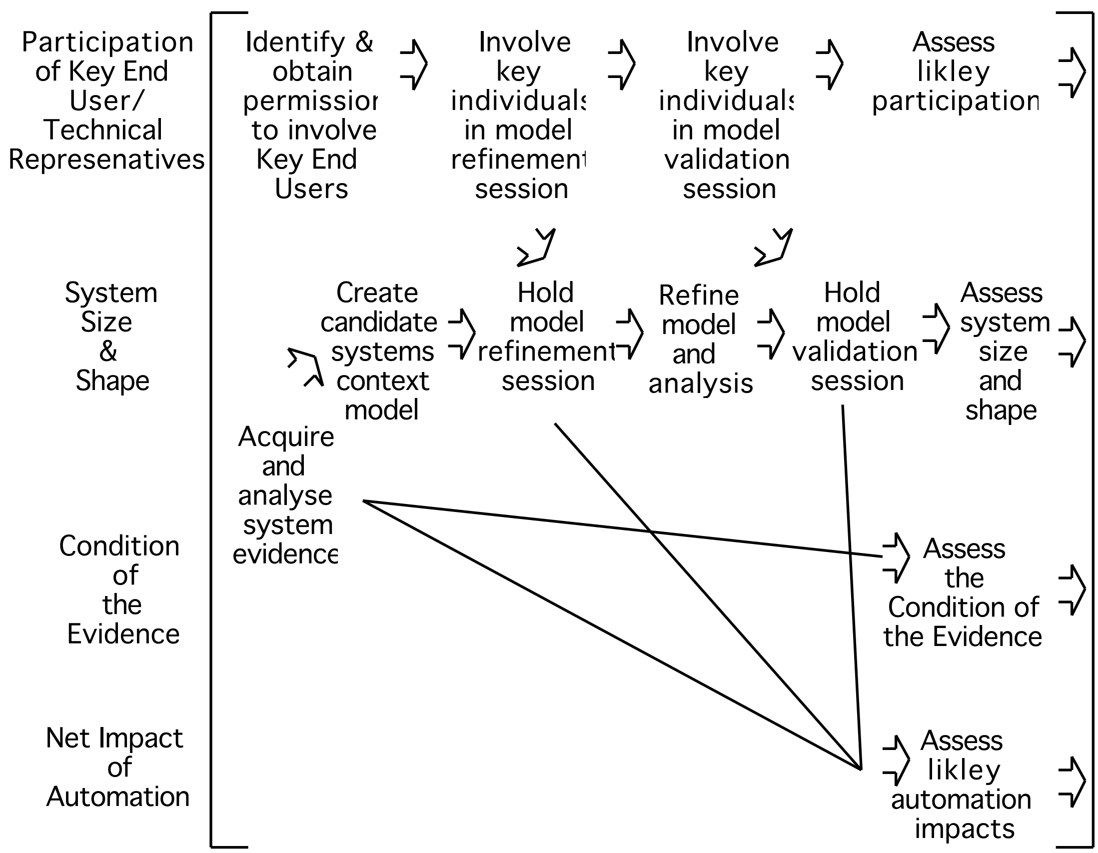

Figure 9 System Survey Overview

1.4 Developing Estimates

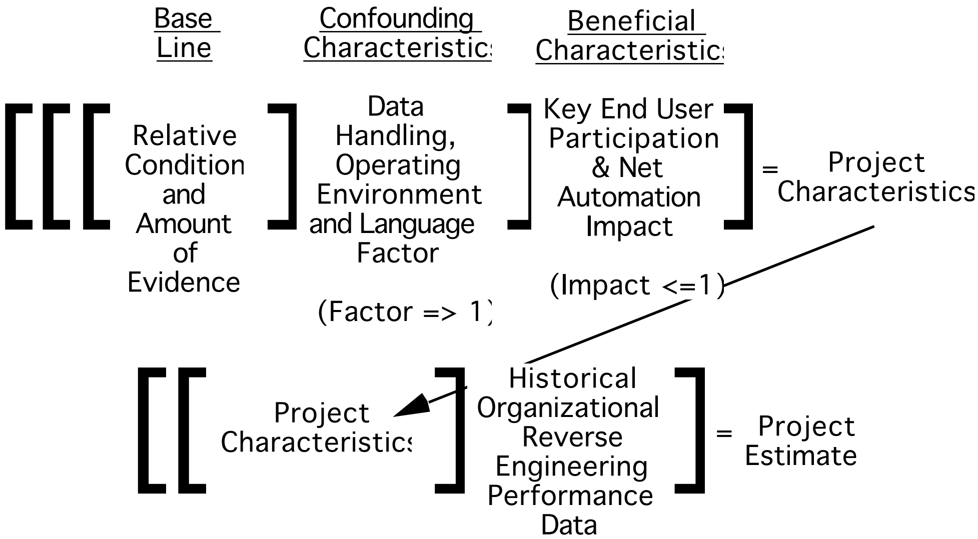

Figure 10 Developing Project Estimates from Project Characteristics

Figure 10 shows how information from the PSS can be combined into components that confound and components that facilitate projects. Recall from Figure 2 the process of combining all of the above information into a project estimate: first assess the project characteristics and then apply organizational productivity data. To obtain a useful assessment as to the project characteristics follow the procedure:

1. Develop a base line assessment of the evidence analysis requirements.
2. Factor in confounding characteristics.
3. Factor in beneficial characteristics.
4. Apply productivity statistics.

Each step is described in the following subsections.

1.4.1 Step 1 - Develop A Base Line Evidence Assessment

Meaningfully, combine the facts gathered describing amount of evidence and the facts describing its relative condition into a baseline data reverse engineering project assessment. This assessment describes the relative size of the analysis job required to analyze the evidence based on its quantity and quality. Analysis of the relative condition and amount of evidence consists of inventorying the number of system data items. Figure 11 shows some of the possible forms of the physical evidence. It also presents the typically available statistics and the desired project planning information. For example, when assessing the task of analyzing the code the typically available statistic is the number of pages of system documentation. The more desirable information desired is the number of hours required to analyze code. But over a series of projects the organization will develop expertise to make these metrics more useful. This results in one or more evidence categories will some measure of the two evidence conditions: synchronization and existence.

	Typically Available Statistic	**Desired Statistic**
Code	Number of pages of code	Number of hours required to analyze code
Files	• Number of files • Number of data items for each file	Number of hours required to analyze files
Reports	• Number of reports • Number of data items on each report	Number of hours required to analyze reports
Screens	• Number of screens • Number of fields on each screen	Number of hours required to analyze screens
Data Structures	• Number of data structures • Number of data items on each data structure	Number of hours required to analyze data structures
Business Rules	Organizational procedures manuals.	Number of hours required to analyze business rules
Documentation	Number of pages of documentation	Number of hours required to analyze system documentation
Total	Total number of data items	Number of hours required to analyze all data items
Overall	Effective data item analysis rate/hour	

Figure 11 Desirable And Typically Available Physical Evidence Statistics

The second baseline component is an assessment of the availability, accessibility, and condition of the system data assets. The goal of this portion of the analysis is an assessment of the condition of the system data assets. This output is usually developed as a by-product of the systems context model development. The assessment is best accomplished by examining the documentation first hand, performing a series of comparisons with the existing system to verify correctness, and following the change history with the documentation changes to get a feel for the system volatility.

This task is a good place to practice some sensitivity because it involves assessing and reporting on the state of the system documentation. System documentation usefulness assessments have sometimes been uncomfortable to professionals who might wish for the resources required to keep it up-to-date. Try to be a lot like Joe Friday, concentrate on the facts with out judging. Future enthusiastic participation of these end users on the performance team is likely to be a key determinant of project success.

1.4.2 Step 2 - Factor In Confounding Characteristics

The confounding effects of system characteristics must be factored into the analysis requirements.

If the date reverse engineering project goals requires an analysis of the source code, the specific programming language requiring analysis can adversely impact the projects. A language factor is assigned according to the class of programming language. Again, the more popular the programming language, the more automated tools supporting data reverse engineering are available. Negative impacts associated with programming languages can come from the difficulties acquiring the assistance of a less popular system or operating environment factor. The degree of platform specific support for the data reverse engineering project can also confound the analysis. Throwing an MVS operating system specialist at a Unix workstation can be confronting to say the least. Working with an unfamiliar operating system environment can adversely effect the productivity. The relative familiarity and popularity of a programming language respectively effect the number of available experts and the general comprehensiveness of the tool support for the language.

Language Category	**Example**
Structured High Order Language	COBOL
Unstructured High Order Language	FORTRAN
Commercial Off The Shelf Assembly Language	Burroughs Assembly
Unstructured Commercial Language	MUMPS
Non-Commercially Supported Language	GWAR

Figure 12 Categories Of System Languages

A data handling system factor can be similarly assigned according to the category of data handling system. Similar to the programming language classification, the more obscure, obsolete or otherwise non-standard the data handling system, the more difficult it has been to access information.

Data Handling System Category	*Factor*	*Example*
Relational DBMS	1	DB2
Hierarchical DBMS	2	IMS
Network	3	IDMS
COTS File Management System	4	VSAM
Non-Commercially Supported System	5	NIH

Figure 13 Categories Of Data Handling System

1.4.3 Step 3 - Factor In Beneficial Characteristics

The beneficial effects of system characteristics must be factored into the analysis requirements.

Factor in project characteristics capable of confounding the analysis such as use of obscure programming languages, operating systems, environments. The net result of this step will be to increase the base line data by some factor greater than one. This project characteristic has components along two dimensions: availability and coverage. End user availability levels are assigned according to the level of participation available - Figure 14.

Availability Level	**Participation**
Full Time	End user are actively participating as a full time member of the performance team. Full time participation usually requires under 20 hours per week.
Half Time	End users are designated as available to answer questions during certain times of the project but are not expected or perhaps needed at all project meetings.
Limited	End users are designated as being able to participate in a limited number of (usually 5) days attendance at data asset refinement/validation sessions. Usually these individuals are in high demand but their presence at the sessions will greatly contribute to the performance team productivity.
Very Limited	End users are in such demand they are available for only abbreviated or partial data asset refinement/validation session attendance.
Not Available	Project feasibility must be re-considered if end users are unavailable.

Figure 14 End-User Availability Levels

Participation requirements for each of the following categories must be defined and the participation of individuals possessing the required knowledge must be secured.

- Effect on productivity due to database expert participation.
- Effect on productivity due to functional participation.
- Effect on productivity due to application program expert participation.
- Net effect on productivity due to domain expert participation.

Factor in project characteristics of capable of reducing the overall resources required to perform the projects. Both the degree of end user participation and the net impact of automation should be capable of reducing the combined product of the base line assessment and any confounding characteristics. An indirect goal of the PSS is to, as much as possible, verify the anticipated participation level of the end users. The importance of obtaining executive sponsorship becomes evident when requesting significant amounts of end user time. The process of holding the two modeling sessions will provide evidence as to any possible discrepancies between proposed and actual availability. Strong attendance and participation at the sessions are more likely indicators of future project success. This step results in the project characteristics.

A consulting organization I know, tells customers that reverse engineering will cost them, *about one dollar for each line of code.* I always wanted to introduce them to the folks I know who program in MUMPS - a programming language, whose programmers who take pride in their ability to write entire programs as single lines of program code [see Aiken et. al. 1994]. This consulting organization is charging to run code through a data reverse engineering tool designed to programmatically extract certain types of information. For certain projects this application of automation will provide rapid access to the required information. If the environment is somewhat standardized (COBOL), the there is more likelihood of automated products supporting the analysis. There are fewer automated tools supporting MUMPS and other less popular system environments. For other data reverse engineering projects, this type of analysis will be infeasible or not useful. Thus automation has the potential to hurt as well as help in data reverse engineering projects. For this reason it is useful to carefully define terms when describing data reverse engineering project goals. Similar to available end user participation levels, automation can also positively effect data reverse engineering project estimates. The goal is to determine the potential technological leverage in light of likely project goals.

The *automation impact* can be defined as the degree to which automated support affects the productivity of either the model validation/refinement sessions or the effectiveness of the off-line analysis. However, since automation can positively effect only on those occasions where project benefits outweigh project costs, it is more useful to speak in terms of a net automation impact. Net automation impact is the amount remaining after the costs of automation are subtracted from the benefits. Examples of possible impacts due to attempts to automate the project are shown in Figure 7-15.

Working on in an unfamiliar operating system environment can adversely effect the productivity. So can working on an antiquated CASE tool or with out them. Specific components of the automation characteristic are presented in Figure 7-14. (The last three items can be offset by the participation of an corresponding end user in the programming language, platform, and data management areas.)

Automation Characteristic Component	**Example**
Weeks of project specific analytic tool learning curve	Having to learn a different CASE Tool because original system models were developed using something other than the current tool.
Weeks of project specific CASE/reverse engineering tool development	Deciding it will be effective to develop a tool specific to a project for example to read and/or analyze non-standard data structures.
Programming Language Effect On Productivity	Systems were developed using obscure programming language or otherwise experiencing a non-availability of language expertise benefit from increased participation from programming language specialists.
Platform/Operating System Effect On Productivity	Projects where systems were developed in obscure platform/operating system environments experience similar relationships between participation and productivity.
Data Management System Effect On Productivity	Projects where systems were developed using popular data management systems can require less expertise than situation where non-standard data management systems were employed.

Figure 15 Automation Characteristic Components

1.4.4 Step 4 - Applying Productivity Statistics

Once obtained the project characteristics can be transformed into an estimate by applying data describing previous organizational reverse engineering productivity. (Until industry averages are established, this information must come from previous organizational data reverse engineering project experiences.) Thus the PSS provides a base line estimate of the number of total data assets to be analyzed in order to be produce the desired project outputs. If the PSS provided indications that the target system maintained information about 200 entities and if your previous data reverse engineering experience indicated the performance teams could define thirty entities during a standard week long modeling session then the total estimated schedule would be a total of 7 data asset refinement/validation sessions or around 28 weeks total project time.

1.5 Additional Considerations and Plans for Future Research

Some of the project variables can only come from your organization. These include the productivity of the data reverse engineering specialists and the cost of end user participation. Also, since the historical data won't exist in organizations that have not performed data reverse engineering projects, it will be difficult to estimate the first project.

A quick word about productivity measurement. The last thing in the world I would suggest is development of quota's with regard to productivity variables. Reverse engineering specialists must collect and report their own data via standard work breakdown structure reporting codes and other existing time management and reporting vehicles. (Throw a redundant data collection device at these individuals and it will definitely get thrown back!) Collect the evidence at post project review sessions by counting the number of data assets created and dividing by the hours allocated for analysis. Over time these numbers can be used to accurately predict this.

In addition, the costs to properly reverse engineer systems and value of the reverse engineering products are consistently under estimated by management. A common misperception is that we just "run the code through a CASE tool and the tool will produce new systems." It has been challenging to convince management that reverse engineering is a substantially broader and more complex task than just "restructuring the code". Should the organization be in a position to, end users have quickly be trained to perform some data engineering functions (i.e., entering project information directly into the project data bank). This form of support can be used to leverage the performance team efforts. It represents an opportunity to save considerable project time and presents a nice opportunity to perform some technology transfer as the end users discover how much easier it can be to maintain CASE-based system documentation.

One obvious extension of this work is to extend this initial proposed framework to include missing components from areas such as database and software reengineering to build .

1.6 References

- Peter Aiken, Alice Muntz, and Russ Richards "DoD Legacy Systems: Reverse Engineering Data Requirements" *Communications of the ACM* May 1994 37(5):26-41.
- Peter Aiken & Pam Piper "Data Reverse Engineering's Role in Enterprise Integration" *Proceedings of the 4th Reengineering Forum*, Victoria, B.C., Canada, September 19-21, 1994.
- Peter Aiken, Michael Joseph & Amita Goyal "Identifying and Evaluationg Data Reverse Engineering Opportunities" *Proceedings of the 4th Reengineering Forum*, Victoria, B.C., Canada, September 19-21, 1994.
- Peter Aiken, "Hypermedia-based Requirements Engineering," chapter in (Stephen J. Andriole, editor) *Advanced Technology for Command and Control Systems Engineering*, AFCEA International, 1991.
- Victor R. Basili "Viewing Maintenance as Reuse-Oriented Software Development" *IEEE Software* January 1990, pp. 19-25.
- Francois Bergeron and Jean-Yves St-Arnaud "Estimation of information systems development efforts," *Information & Management*, 22 (1992), 239-254.
- Fredrick P. Brooks, Jr. *The mythical man-month; essays on software engineering* Reading, Mass., Addison-Wesley, 1975.
- Canfora, G.; Cimitile A. "Reverse-engineering and intermodular data flow: a theoretical approach" *Journal of Sofware Maintenance: Research and Practice* Vol: 4 Iss: 1 p. 37-59 March 1992
- Elliot Chikofsky & James H. Cross II "Reverse Engineering and Design Recovery: A Taxonomy" *IEEE Software* January 1990 7(1):13-17.
- Elliot Chikofsky "The database as a business road map" *Database Programming and Design* May 90 3(5):62-67.
- Christman, R.D.; Cook, D.P.; Mercier, C.W. "Reengineering the Los Alamos common file system" *Digest of Papers. Tenth IEEE Symposium on Mass Storage Systems. Crisis in Mass Storage* (Cat. No.90CH2844-9) p. 122-5 Editors: Friedman, K.D.; O'Lear, B.T. IEEE Comput. Soc. Press Washington, DC, 1990
- Duncan G. Connall & Daniel R. Burns "Reverse Engineering: Getting a Grip on Legacy Systems" *Data Management Review* October 1993, pp. 24-27.
- Delligatta, A. "System reengineering and the user" *Information Systems Management* Vol: 9 Iss: 1 p. 76-7 Winter 1992
- Hall "Overview of reverse engineering and reuse research" *Information and Software Technology* Vol: 34 Iss: 4 p. 239-49 April 1992
- Capers Jones *Assessment and Control of Software Risks* Englewood Cliffs, N.J. : Yourdon Press, 1994.
- Thomas J. McCabe "A Complexity Measure" *IEEE Transactions on Software Engineering* December 1976 SE-2(4):308-320.
- Q. Ning "User involvement in software re-engineering" *Fourth International Workshop on Computer-Aided Software Engineering. Advance Working Papers* (Cat. No.90CH2956-1) p. 28-9 Editors: Norman, R.J.; Van Ghent, R. IEEE Comput. Soc. Press Los Alamitos, CA, USA 1990.
- A. Ricketts, J. C. DelMonaaco & M. M. Weeks "Data Reengineering for Application Systems" *Proc. Conference on Software Maintenance-1989* IEEE Computer Society Press pp. 174-179.

- M. Sneed "Economics of software re-engineering" *Journal of Sofware Maintenance: Research and Practice* Vol: 3 Iss: 3 p. 163-82 Sept. 1991
- William M. Ulrich "From Legacy Systems to Strategic Architectures" *Software Engineering Strategies* March/April 1994, 2(1):18-30.
- Van Kirk, Doug "Downsizing host applications; data conversion is often the last but most critical step in re-engineering. (Enterprise Computing)" *InfoWorld* Dec 28 1992 v14 n52-1 p39(2)

1.7 Communication