# *A Method for Assessing the Development Process of Small Organizations*

*Joel Henry, Susan L. Keenen, Michael A. Keenen and Sallie Henry*

**TR 92-49**

Department of Computer Science
Virginia Polytechnic Institute and State University
Blacksburg, Virginia 24061

September 7, 1992

# A METHOD FOR ASSESSING THE DEVELOPMENT PROCESS OF SMALL ORGANIZATIONS

Joel Henry, Susan L. Keenan, Michael A. Keenan and Sallie Henry

Virginia Polytechnic Institute and State University, Blacksburg Virginia

**Abstract** - *This paper describes a two-phase method for assessing the software development process of small organizations. The initial phase involves using the SEI assessment method to establish the activities comprising the development process. The second phase consists of follow-up interviews to discover the implementation technique for each existing activity and investigate the effectiveness of the techniques. Procedures for generating recommendations are outlined. The data acquired is statistically valid and allows discovery of significant problem areas. Recommendations based on the Software Engineering Institute(SEI) framework and accepted software engineering practices are described. The assessment method is evaluated and future work is outlined.*

## Introduction.

The software development process is recognized as having significant impact on the quality of the resulting product. In response to this recognition, a process assessment procedure based on a five level maturity framework has been developed by the Software Engineering Institute[HUMW87]. The maturity framework and associated assessment procedure is the focus of much attention throughout the research community. Additionally, the software capability evaluation(SCE), based on the maturity framework, is included in the evaluation criteria of contractors bidding for certain Department of Defense contracts[MOGJ90].

While the SEI software capability evaluation represents a significant advance in the evaluation of the development process, it has not been free of criticism[BOLT91]. Early in our research the authors recognized three important problems concerning the assessment procedure. These are:

1. Evaluation of a development process is based on the existence of a process structure at one of the five defined maturity levels.

2. The assessment procedure specifically targets large development organizations.

3. No guidelines exist for consideration of organization specific development factors.

We felt the SEI assessment could be augmented with a second phase which evaluated an organization based on the objectives of the five maturity levels. The second phase includes consideration of financial and staffing limitations as well as the specific commercial mission of an organization.

## Problem Discussion.

An assessment of any software organization should be viewed as an investigation into how that organization implements accepted software engineering practices. The process structure upon which the SEI maturity model is based is not applicable to the software industry as a whole[BOLT91]. Many different process structures meet the goals of the SEI maturity framework but would fail an SCE. Our assessment method investigates an organization based on the objectives of the maturity framework, not the process structure underlying the SEI assessment.

The SEI assessment procedure specifically targets large development organizations. Many of the requirements of maturity levels above level 2 require functional groups of dedicated personnel(such as Software Quality Assurance, Software Engineering Process Group and Configuration Management). The functionality of these groups are critical to the SEI maturity framework but separate groups are not required to provide this functionality. Our assessment method investigates how this functionality is achieved within small organizations but does not require separate functional groups.

Software development organizations have very different commercial ambitions within the software industry. Application areas, development goals, and competition make many organizations so vastly different that any assessment procedure not considering these issues overlooks significant organizational factors. The assessment method we propose integrates these factors into both the evaluation and improvement recommendations.

## The Assessment Method.

There are three primary objectives of the assessment method. First, a determination is made of the the process activities currently performed by an organization. Second, the activities that are and integral part of the development process are investigated. The assessment also investigates the methods and techniques used to implement those activities. The data collected is used to isolate high-priority problem areas within a particular software process. Third, specific corrective recommendations are made based for attacking problem areas.

Since the method targets small companies, it is necessary to minimize the use of staff time. The assessment must also guarantee internal and external confidentiality. Internal confidentiality refers to guarding the individual source of

assessment data. External confidentiality refers to guarding the assessment results from the general public. These two issues are particularly important to organizations and to the ability to effectively assess process and produce valuable results.

The first step in the assessment method is to determine the staff members to be interviewed. This is done using an initial background questionnaire to determine the whether a staff member is directly involved in the software development process, is only tangentially involved, or is not involved at all.

Secondly, we administer the SEI assessment questionnaire and analyze the results to establish the existence of specific activities within the process. However, instead of using the follow-up questions from the SEI procedure we conducted follow-up interviews with the development staff. The purpose of the interviews is to establish how and by whom process activities are performed. The follow-up interviews are guided by our own questionnaire. The questionnaire we use investigates activities in a top-down fashion, gathering specific process data after discussion of the process at a high-level. The selection of questions is based on the results of the SEI questionnaire.

The third step is to analyze the follow-up data to determine the techniques used to implement process activities. During this stage we also identify the most significant problem areas based on collected data and substantiated software engineering objectives, principles and attributes[ARTJ87]. The analysis step carefully considers organizational factors as well as assessment data.

Finally, solutions are recommended based on the SEI framework and accepted software engineering practices. The recommendations are given via an oral presentation as well as in written report form.

**Assessment Results.**

Three very different small software development organizations were assessed(referred to as A, B and C). In order to put the recommendations made to each organization in perspective, each firm's background will be briefly discussed. The assessment findings based on analysis of the collected data are presented. Recommendations are given for high-priority problem areas and the implementation approach for the recommendations are summarized.

**Organization A.**

Organization A typically develops software products of between 50,000 and 80,000 source lines of code. Functional prototyping is used to determine and specify user requirements for the target software system. Organization A rarely develops the final software product from it's prototype. Users occasionally opt to use the prototype in lieu of having the final production system built. This is not known during the prototyping phase. As a result, internal documentation and written specifications are often provided after the functional capability of the intended system is accepted. Only in-house testing is performed to ensure validation based on desired functionality and not documented system requirements.

Based on the initial questionnaires, Organization A was classified at Level 1 of the SEI process maturity framework. In addition, statistically significant discrepancies were found to exist between manager and programmer responses in both the initial questionnaire and follow-up interviews. The discrepancies existed in five areas:

- availability of adequate development tools
- effort expended toward developing reusable code
- degree of control over changes made to the software design
- the number of formal test case reviews conducted
- the prototyping methods used in designing the man-machine interface

The breadth of software engineering topics spanned by these five areas was an early indication of the need for a more clearly defined software process. The need for a well-defined process was further substantiated during the follow-up interviews as the development staff voiced concern in areas directly related to process definition. For example, many staff members complained of lack of organization, confusion over individual task scope and misunderstanding the roles of co-workers.

Other results included a lack of code walkthroughs, an absence of the collection of process data (size, time, manpower and number of defects detected) during each project, inadequate version control and lack of detailed specifications.

All findings listed above would typically be viewed negatively in a process assessment. However, the lack of detailed specification is expected in a prototyping environment. Defect prevention is also normally minimized to facilitate rapid development. This data is consistent with the organizational mission of Organization A.

The following recommendations for solutions to the high priority problems found were made to Organization A with respect to it's size, number of part-time employees, it's ability to absorb implementation costs and organizational factors:

1. Produce design documentation.
2. Define the development process in the following areas:
   a. establish change control procedures
   b. plan for better version control
   c. establish and enforce coding standards
   d. conduct defect prevention on selected, critical system elements
3. Collect and maintain a database of process data.

Prototyping typically involves the relaxation of documentation and error handling, among other production

quality systems. The fact some prototypes are accepted as the final product makes design documentation a necessary activity. Design documentation is also beneficial because of the iterations involved in prototyping.

Definition of a process structure with control over the activities involved in prototype development will improve both the quality and speed with which the prototypes are produced. Although prototyping encourages the use of development iterations, adopting a well-structured approach will help control the use of available human and financial resources by pinpointing areas for improvement.

Change control and versioning are critical for Organization A because of the large number of part-time employees involved in product development.

The establishment of coding standards provides the necessary framework for the performance of quality assurance activities as well as establishing a single coding approach.

While extensive defect prevention is inappropriate in a prototyping environment, defect prevention on critical system components is valuable.

Implementation of these suggestions were recommended in two ways. First the functions of a Software Quality Assurance Group could be provided for individual projects using developers and managers from other projects. These individuals would devote only a portion of their time to quality assurance and would report to the project manager.

Secondly, the addition of one new employee to the current group of managers would permit the new employee to oversee all process activities and reduce new responsibilities allocated to the staff performing project quality assurance.

### Organization B.

Organization B primarily maintains a single product with a relatively long history. The product performs information storage and retrieval of large amounts of data. The application system is highly user-interactive with a world-wide customer base. The focus of Organization B is primarily to maintain and adapt a single product to an increasingly diverse customer community.

The results obtained from the initial questionnaires at place Organization B at Level 1 in the SEI process maturity framework. Responses at this phase of the assessment indicated some advanced techniques of a well-structure software process existed. These included the use of individual workstations, a defined method to control changes and effective managerial control of development projects.

There were no statistical differences between the responses given by the programmers and managers (in fact, a positive linear correlation existed).

Problem areas discovered during the assessment included:

- attendance at training sessions was not encouraged
- standards for design and code are not used
- some process and product data is collected but no organized attempt is made to use this data
- confusion exists over who is responsible for quality
- software engineers consistently complained about the lack of communication in general
- management was responsible for the majority of design, testing, technical decisions, quality assurance and control, configuration management as well as project management

The problem areas within the development process of Organization B were analyzed considering organizational factors. In this case the organizational factors were not used to explain the assessment results. Rather, the organizational emphasis on maintenance and adaptation directed corrective recommendations in a direction most beneficial to these activities.

The problem areas indicated weak implementation of the principles of concurrent documentation and life-cycle verification[ARTJ87]. In addition we discovered the large number of tasks performed by management were not performed effectively. Management was simply responsible for too many functions to be effective at each.

The following recommendations were made to Organization B:

1. Document the development and maintenance process.
2. Develop module history folders containing design, defect and change information.
3. Establish and enforce coding standards.
4. Implement standard unit and integration testing procedures.
5. Designate one member of each project group as a librarian who performs configuration management functions.
6. Form a quality assurance group consisting development staff performing assurance functions part-time.

Documentation of the development process provides communication to software engineers and material for more effective training session.

The development and use of module history folders and coding standards are two more practices critical for maintenance and for communication.

Standard testing procedures are beneficial for consistent defect detection. They become training materials as well as allowing staff members to conduct testing now being performed by management.

A project librarian further reduces the number of tasks currently performed by management.

Again we recommended the creation of Software Quality Assurance Group involving developers and managers from other projects. These individuals would devote only a portion of their time to quality assurance and would report to the project manager.

### Organization C.

Organization C produces and maintains a complex real-time software system for a network of distributed hardware. Two smaller software products are also developed and maintained at Organization A.

The organization began with a small group of extremely talented individuals. These individuals produced a software system of more than 300,000 source lines of code within just a few months in order to meet a market deadline. Delay in the delivery of this product would have resulted in the complete failure of Organization A. Many basic software engineering practices were not implemented because of the tremendous schedule pressure.

The results obtained from the initial questionnaires at Organization C place the firm at Level 1 in the SEI maturity framework. Data obtained in this phase of the assessment indicated some advanced techniques of a well-structure software process existed. These included the use of individual workstations, an effective customer-developer interface and effective technical control of development projects.

We quickly recognized during initial data collection that a project manager was missing for the largest software product and that two staff members we termed 'recognized leaders', assumed some management responsibilities.

There was a significant statistical difference between the responses given by the software engineers and managers in the area of process definition. The software engineers perceived no defined process while management and the 'recognized leaders' steadfastly maintained a well-defined process existed.

Other problem areas discovered during the assessment included:

- testing is not planned
- testing time is less than one-third coding time
- coding standards are not enforced
- design documentation is lacking
- employee acclimation is six to twelve months
- more than 70% of the maintenance man-hours are spent on less than 35% of the source code
- system testing is not performed prior to delivery

Many of the problem areas within the development process of Organization C are the result of the schedule pressure under which the original software product was developed.

The lack of a project manager, responsible for defining and changing the software process, project tracking, enforcing standards, as well as other duties is clearly a critical problem.

Testing is done by individual programmers on an ad hoc basis. There is no evidence of a standard testing method and testing effort appears to be alarmingly short.

A lack of formal documentation procedures pervades the development process. This is the cause of the high acclimation period and difference of view regarding the existence of a defined process.

Further investigation into the cause for such a large expenditure of maintenance effort on a small amount of source code uncovered interesting information during our follow-up interviews. The source code requiring a large maintenance effort was produced by a member of the original development group. This individual is no longer with the organization. The lack of formal documentation during initial development is responsible for maintenance overhead on a small portion of code.

The lack of complete system testing was initially identified as a problem area. Investigation into this perceived problem area yielded provided qualifying information. System testing is impossible due to the size and distribution of hardware controlled by the software system. It is simply impossible to create a network of thousands of machines over hundreds of square miles in order to test the system before delivery. Alternatively, Organization C has gone to great lengths to simulate such a hardware system in order to test the software product in-house. While full-blown system testing is more desirable, this is the only alternative available within the cost, time and manpower constraints of Organization C.

The importance of the largest software product to the commercial success of Organization C needed to be considered in formulating our recommendations. No large software products are to be developed in the near future. For these two reasons our assessment recommendations targeted maintenance and testing issues.

The following recommendations were made to Organization C:

1. Document the design structure of the existing software products.
2. Implement standard unit and integration testing procedures.
3. Establish an organizational commitment to testing.
4. Document the development process.
5. Specify the responsibilities of a project manager, create the position and select a project manager for the large software product effort.

Design documentation is critical for maintenance and for the reduction of employee acclimation time. The large effort spent on a small portion of the source code results from the lack of design documentation. This problem could become

overwhelming if other members of the original development group left Organization C.

Establishing an organizational commitment to testing involves the development of standard unit and integration testing procedures as well as focusing on testing organization-wide. These two actions will start Organization C toward effective, consistent testing.

Documenting the development process forces management and the 'recognized leaders' to produce a model of the defined process they claim exists. Documentation of the development process can then be used to clarify staff roles, analyze the development process and reduce the long employee acclimation period.

Project management is a critical requirement of the SEI process maturity framework. Organizations large or small require effective project management. Organization C must specify the responsibilities of project manager, create a project management position for the large software product and fill this position with qualified person. Software project management is critical for planning, progress assessment, staff coordination and process execution.

**Evaluation of the Assessment Method.**

Since it is our hypothesis that small organizations by their very nature will be classified as Level 1 in the SEI hierarchy, we feel process data as well as organizational factors must be considered.

Process data pertains to the activities within a software development process. The quality of the software products produced by most small organizations must be the result of the application of sound software engineering practices applied at some level. Our assessment method attempts to how and by whom these practices are applied.

Our use of the SEI assessment questionnaire allowed us to establish the existence of activities within the development process of the organizations. Also, we were able to discover the techniques used to implement those activities. By using the SEI instrument, we could statistically test for certain correlations or contradictions within the sampled data. This allowed us to perform in-depth, follow-up interviews with the development staff.

By taking organizational factors into account, our assessment method is flexible enough to evaluate the software development process of three small organizations with very different software development functions.

We are able to make recommendations that are model-based rather than question-based[MOGJ90]. Model-based process improvement augments an existing process with cost-effective solutions of high-priority problem area. Question-based process improvement improves a process by providing positive responses to assessment questions. We are strongly against this approach to process improvement.

The assessment method had little impact on the operations of the evaluated organizations. Data acquisition requires no more than two hours of time per staff member. We actually spent considerably more than two hours with a number of the development staff at their request.

All three organizations were enthusiastic about and supportive of the assessment. The development staff of each organization contributed their ideas and impressions generously. There seems to be a genuine interest among many small organizations to be evaluated and receive recommendations for improving their software development process. One author has received requests for assessment from two competitors of Organization C.

We must also point out that our assessment technique has certain weaknesses. Among these is the necessity to use subjective judgement in order to evaluate the effectiveness of an organization's techniques.

Our recommendations, while based on accepted software engineering practices, are not able to make use of proven implementation techniques for small organizations. This type of foundation does not exist at this time.

The reliance upon the SEI questionnaire to do an initial evaluation is subject to question since the questionnaire is not intended to assess small organizations. While our use of the SEI questionnaire is not evaluatory, it is intended to establish the existence of activities within the development process. The 101 questions on the questionnaire are far more than are needed to provide activity existence information. A more efficient technique for satisfying this goal need be developed.

This last observations raises another issue that should be made about our assessment method. It should be noted that all three organizations are classified as Level 1 by means of the SEI evaluation procedure; however, there is a vast difference among the processes at those organizations. It has been suggested that the SEI procedure is not precise enough to make distinctions among various Level 1 organizations[BOLT91]. It is the opinion of the authors that there should be refinements of the maturity classification within Level 1, 2 and 3 in the SEI scheme.

**Future Work.**

Much work remains to be done to improve and extend our assessment method. The use of the SEI questionnaire in the initial phase of the assessment could be refined based on our results. The purpose of the questionnaire is to launch an assessment. Questions specific to software engineering practices and principles as well as organizational factors would be more helpful.

It is our view that multiple grades of maturity exist within SEI maturity levels 1, 2 and 3. Our results support this view as we found differences between all three organizations participating in our assessment despite their common score of maturity level 1. In addition an organization may be

more mature in certain aspects of the development process or within particular phases. This information should be more definitively included in an assessment.

The top-down approach of questioning specific software engineering practices and principles allows pertinent data to be included in the evaluation. More extensive data points should be developed.

## Conclusions.

We have proposed and implemented an assessment method for small organizations utilizing the objectives and practices of the SEI maturity framework. The assessment method does not rely on a single process structure. The method considers financial and staffing constraints as well as the commercial mission of small organizations during assessment. Recommendations for process improvement are made using best software engineering practices and organization specific factors.

## References


[ARTJ87] Arthur, J.D., Nance, R.E., and Henry, S.M., *A Procedural Approach to Evaluating Software Development Methodologies: The Foundation*, Technical Report TR 86-24, Virginia Tech, Blacksburg, Virginia, 1987.

[BOLT91] Bollinger, T.B. and McGowen, C., "A Critical Look at Software Capability Evaluations," *IEEE Software*, July 1991.

[HUMW87] Humphrey, W.S. and Sweet, W.L. "A Method for Assessing the Software Engineering Capability of Contractors," Software Engineering Institute, Carnegie Mellon University, September 1987.

[HUMW89] Humphrey, W.S., *Managing the Software Process*, Addison-Wesley, 1989.

[HUMW91] Humphrey, W.S. and Curtis, B., "Comments on 'A Critical Look'," *IEEE Software*, July 1991.

[MOGJ90] Mogilensky, J., "Approaches to Upgrading Software Process Maturity," *Washington Ada Symposium*, June 1990.