Integrating TQM and SEI Process Assessment

Joel Henry and Sallie Henry

TR 92-27

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

May 19, 1992

Integrating TQM and SEI Process Assessment

Joel Henry and Sallie Henry
Department of Computer Science
Virginia Tech
Blacksburg VA 24060
Phone: 703-552-3208

Abstract: This paper describes a methodology for assessing the software process (both development and maintenance) used by an organization. The assessment methodology integrates the principles of Total Quality Management and the work of the Software Engineering Institute. Assessment results in a well-understood, well-documented, quantitatively evaluated software process. The methodology utilizes four steps: investigation, modeling, data collection, and analysis of both process content and process output. The investigation step of the methodology gathers information about the activities used by an organization and the environmental factors affecting the process. The modeling step produces a graphical representation of the activities comprising the process. Data collection gathers quantitative process data and post-delivery data. The analysis step reveals problem areas within the process through statistical analysis and process content analysis. Process improvements are determined by analysis results.

1 Introduction

The development and maintenance of software has yet to employ a body of scientific knowledge and a measurable set of engineering practices similar to those used by such disciplines as Electrical Engineering, Mechanical Engineering and Chemical Engineering. Software Engineering as a discipline is struggling through the early stages of technical development. While statements such as "The software crisis is dead" have been made [HUMW89], a realistic view of the current state of the practice places Software Engineering early in the commercial stage of technical development, as shown in Figure 1. Software Engineering is a craft becoming commercial.

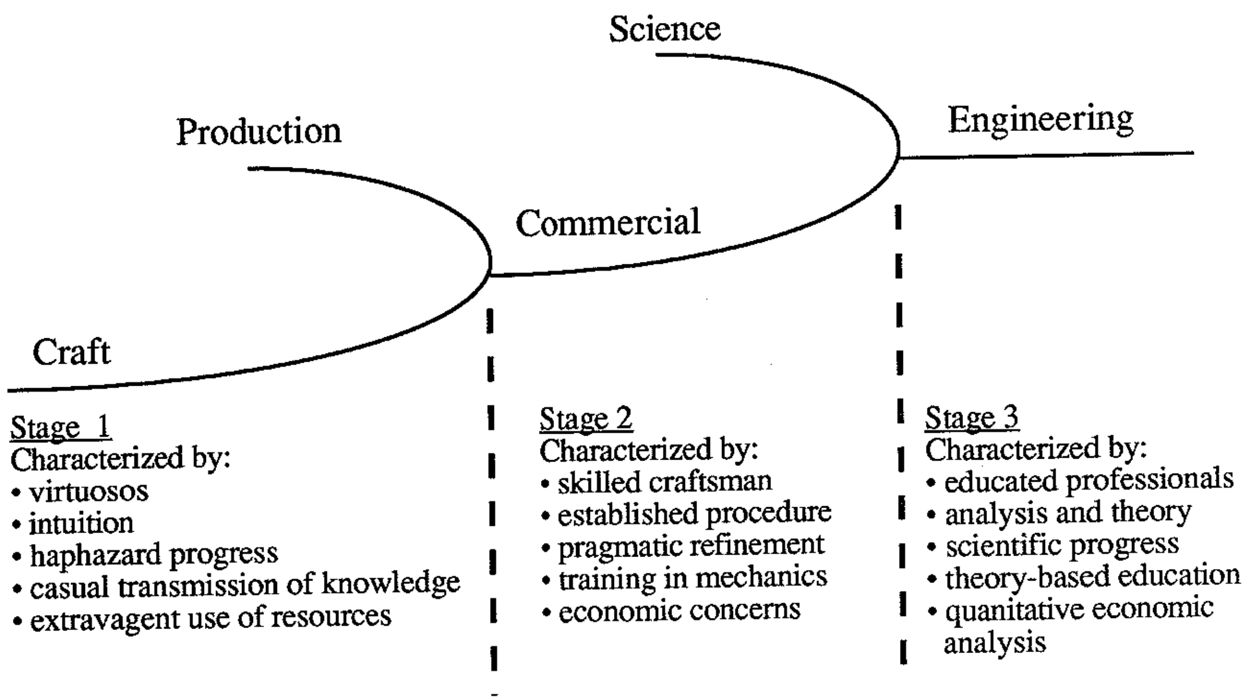

Figure 1. Stages of Technical Development

The need remains for the consistent production and effective maintenance of high quality software, on time and within budget. In a time of reduced budgets and increased competition for software contracts organizations must improve the quality of the software products they produce and maintain while reducing cost.

The software process, including both development and maintenance, significantly impacts the quality, timeliness and cost of the software products produced [HUMW87] [HUMW89] [MOGJ90] [OSTL87] [BOLT91] [CRAS85] [GESD91] [HUM91]. Arthur et. al. substantiate a linkage of process to product quality through the use of software engineering principles such as life cycle verification, early error detection and concurrent documentation [ARTJ86]. This work argues that quality attributes are imparted to software products through the effective application of the appropriate principles. Product cost and timeliness are also largely determined by the software process through software engineering management practices [QUIR80]. Huff et. al. have shown the affect of the software process on product cost and delivery schedule through the management practices employed by the process [HUFK86].

2 Software Process Assessment and Improvement

Assessing and improving the software process are difficult tasks. Process assessment results in a well-understood, well-documented and quantitatively evaluated software process. Only in a well-understood process can the impact of an improvement to be fully measured. Process improvements are more easily implemented, and more permanent, when the process is documented. Quantitative evaluation of the process supports objective feedback on the benefit provided by process improvement.

The Software Engineering Institute(SEI) has been in the forefront of software process research during the past five years, producing a process maturity framework, a capability assessment procedure and a capability maturity model (currently under review, scheduled for release in late 1992) [HUMW87] [SEI91]. SEI efforts have drawn both praise and criticism [BOLT91] [HUMW91] [MOGJ90]. While debate over the SEI approach continues, many experts in the field of Software Engineering agree that the greatest advancements in the field will result from assessing and improving the software process.

The two most important approaches to software process assessment and improvement are the SEI Contractor Capability Assessment and the principles of Total Quality Management(TQM) [SEIS92]. While the goals of both approaches are to improve product quality by improving the process, the methods used are quite different.

2.1 SEI Process Assessment and Improvement

SEI Contractor Capability Assessment satisfies the need to discover and understand problems within an organization's software process. The assessment is a balance between an unconstrained search for problems and the urge to prematurely specify solutions.

The objectives of SEI process assessment are to:

- learn how the organization software process works, and
- identify problem areas within the process [HUMW89].

The SEI Contractor Capability Assessment method is based on a five-level process maturity framework. The process maturity levels are:

1. Initial - Ill-defined procedures and controls, result in little consistency in managing the software process.

2. Repeatable - An organized and stable software process results in the ability to measure and trace the process and produce accurate estimates of costs and schedules.

3. Defined - A well-defined software process and the collection of tracking data allow verification of compliance with designated standards.

4. Managed - Analysis of quantitative measures of all aspects of the software process results in control of the process.

5. Optimized - Control of the software process provides an opportunity to modify and improve the process using process data.

The process maturity levels form a logical progression of more sophisticated process descriptions. The activities required in each level are drawn from accepted software engineering practices and proven industrial engineering techniques.

SEI process assessments involve the following steps:

1. Secure management commitment to change prior to assessment.
2. Administer the questionnaire and collect the results.
3. Conduct follow-up interviews to substantiate the results of the questionnaire.
4. Give a preliminary presentation of the assessment results.
5. Produce a thorough written report of the assessment results.

The SEI assessment method begins by receiving management commitment to the assessment and subsequent improvement. Without such commitment assessment is difficult if not impossible and subsequent improvements have little chance of implementation.

The second and third steps gather substantiated answers to 101 yes/no questions. The questions are separated into groups corresponding to process maturity levels 2 through 5. The purpose of each group is to evaluate the organization's process against the required activities for a specific maturity level.

Questionnaire and follow-up data is then used to assess the software process used by an organization, labeling the organizations process as maturity level 1, 2, 3, 4, or 5. Process improvement is achieved by adding the activities missing from an organization's process, but required by the SEI Process Maturity Level.

The SEI process assessment is successful in gathering data which can be statistically tested [HENJ91a]. The presence or absence of the activities required to achieve a specific maturity level is discovered. However, the SEI method does not detect all the activities used by an organization. Activities not specified within the Process Maturity Framework are not detected [BOLT91] [HENJ91a]. In addition the assessment method does not specify how to tailor assessment results to an organization. Organizational goals and environmental factors are not considered in the associated assessment, consequently this type of information is not captured. Bolten and McGowen assert that much more information is elicited than captured in the investigation method [BOLT91].

2.2 TQM Process Assessment and Improvement

TQM is a general approach to product quality improvement. TQM is defined as "A cooperative way of doing business that relies on the talents and capabilities of both management and labor to continually improve quality and productivity using teams." [JABJ91] The six principles of TQM are as follows:

1. Maintain customer focus.
2. Focus on the process.
3. Prevent of defects rather than inspect for defects.
4. Utilize the knowledge and expertise of labor.
5. Perform fact-based decision making.
6. Integrate feedback to continuously improve the process.

These general purpose principles are applicable to nearly all types of industry, and particularly to the commercial production of software. The emphasis on process assessment and improvement reflects the belief that quality products result from a quality process.

TQM is implemented in five phases:

- preparation,
- planning,
- assessment,
- implementation, and
- diversification.

The assessment and implementation phases of TQM are considered in the integrated methodology. Organizational assessment is typically performed through self-evaluation, customer surveys or training feedback. No single structured approach to organizational assessment is specified in TQM.

The implementation phase requires process definition, measurement and improvement based on statistical analysis [DOD88]. Process definition is intended to document the process, defining roles and clarifying how the process is actually performed. Measurement captures defect data about the product at various points during the process. Control charts are established for statistical quality control. These control charts show the average number of product defects detected following each activity and control limits. The control limits depict acceptable deviations from the average and are determined using statistical techniques. If the number of product defects exceeds control limits the process is said to be out of control and management is alerted. Process improvement is achieved by reducing the average number of product defects.

The strengths of TQM are reliance on objective measurement, the application of statistical quality control, the commitment of all personnel to quality throughout the organization [DOD88]. The application of TQM to the software process is not clearly defined. TQM is intended as a general approach to manufacturing process improvement, consequently there are no requirements for the presence of specific activities within a process. The activities used in a manufacturing environment are better understood(activities such as welding, soldering, painting, etc.) and involve physical components. Software process activities are not as well developed and operate on concepts and abstractions. Exactly which activities a given organization should include in their software process is not clear.

3 Integrated Process Assessment

The SEI and TQM approaches differ significantly The SEI approach can be viewed as a process content assessment while TQM can be viewed as a statistical data assessment. SEI process assessment examines activities within the software process in detail, adding activities or altering the flow of control between activities. TQM also considers the activities within the process but concentrates on improving product measures following each task.

The assessment methodology outlined in this paper integrates concepts from both the SEI approach and the TQM principles. The basis for integrating these two approaches is:

- The SEI assessment specifies the activities comprising the software process. The TQM implementation phase evaluates the effectiveness of the activities.

The specific goals of this assessment methodology are:

1. To discover the activities comprising an organizations software process and the environmental factors affecting the process.
2. To graphically document the process in a form usable by software personnel and amenable to analysis.
3. To gather quantitative process and post-delivery product data.
4. To analyze an organizations process by considering activities present and absent within the process as well as statistical relationships between and among quantitative data.
5. To determine what process improvements are needed.

The software process assessment methodology outlined in this paper achieves these goals using four steps:

1. Investigate the process.
2. Model the process.
3. Gather data.
4. Analyze the data.

This paper describes a four step software process assessment methodology. The assessment methodology investigates, documents, instruments and quantitatively evaluates a software process. The investigation step of the methodology determines the software process used by an organization and the environmental factors affecting the process. Investigation is the basis for process documentation in the form of a process model. The process is instrumented based on the model and investigation results. Quantitative process assessment is performed using statistical techniques.

The methodology utilizes both the SEI approach and the TQM approach to process assessment. Investigation and modeling discover and document the activities used by an organization. Data gathering and statistical analysis validate relationships and influences throughout the software process. Consideration of the activities within the process and statistical relationships are needed to determine what types of improvements are needed and how to implement the improvements.

3.1 The Investigation Method

An effective process investigation method captures both objective and subjective data. A well-structured investigation obtains data from organizational, functional and behavioral views. Integration of SEI techniques, process audits and modeling views results in a powerful, multi-step investigation method [HENJ91a]. The process investigation method included in this assessment methodology integrates these three areas of process research.

The process investigation method uses the SEI questionnaire, obtaining data which can be analyzed using statistical techniques. The results of the questionnaire are used in a much different way. Rather than analyzing questionnaire results to determine what answers must be substantiated, the results are used to develop more general questions used in the follow-up interview. The follow-up questions are created by consideration of process maturity framework goals, software engineering principles and accepted software management practices. These questions investigate the process in a top-down fashion, discovering the activities used in the software process, regardless of their inclusion in the SEI process maturity framework.

The three primary objectives of the investigation method are:

1. To determine the software process used by an organization.
2. To discover the significant factors affecting an organizations software process.
3. To define the methods and techniques used to implement the activities comprising the process.

There are four specific phases to the process investigation method. The first phase in the investigation method determines the staff members to be interviewed. This is done using an initial background questionnaire establishing whether each staff member is directly involved in the software process, only tangentially involved, or not involved. The SEI assessment questionnaire is used in the second phase to establish the existence of specific activities within the process. The third phase requires follow-up interviews be conducted using more general questions. The purpose of the interviews is to discover activities not included in the SEI Process Maturity Framework. In addition, follow-up interviews establish how and by whom process activities are performed. The fourth phase analyzes the follow-up data to determine the differing views of the process within the organization, environmental factors affecting the process and other process activity information. The analysis phase carefully considers organizational factors as well as investigation information to form a correct, complete view of the software process used by an organization.

This investigation method has been successfully performed with excellent results [HENJ91a]. Information about an organization's software process is captured and important problem areas discovered. Many of the recommended improvements have been implemented [HENJ91a] [BOYJ91].

3.2 The Modeling Technique

Once a software process is understood, it must be documented to insure consistent use and effective process improvement. Documenting a software process typically results in clarification of the process and the roles of both individuals and groups within the organization. A documented process also aids in establishing what type of data to collect, where to collect the data and what the collected data represent. Process documentation, in the form of a model, is critical to process assessment and improvement.

The modeling technique is based on control flow diagrams. Control flow diagrams are well suited to modeling software processes because they clearly depict the interactions among process activities. Control flow models of a process are constructed according to specific rules based on the concepts of top-down functional decomposition, information hiding and stepwise refinement.

Development of the modeling technique involved four major steps, each intended to impart some of the desired characteristics to the resulting models. First, it was determined that four tiers, each with increasing amounts of detail, would be used to construct process models. Secondly, the purpose of each tier was defined. Thirdly, model traceability constraints were specified to insure consistent representation of a process between different tiers. Finally, rules governing the generation of a model through all tiers, and specific rules for the generation of each tier were defined.

The four tiers used in this modeling technique are shown in Figure 2 and described as follows:

1. Tier 1 - Phase Tier. The purpose of Tier 1 is to name each phase within the process. For each phase, the products input and output, interphase communication, measurements and the group responsible for implementing the phase are specified.

2. Tier 2 - Task Tier. The purpose of this tier is to describe what major tasks need to be accomplished to implement a phase. A separate task tier exists for each phase.

3. Tier 3 - Procedure Tier. The purpose of this tier is to specify how the major tasks specified in Tier 2 are performed. The major implementation steps for each task shown in Tier 2 are described in Tier 3. A separate procedure tier exists for each phase.

4. Tier 4 - Procedure Step Tier. The purpose of this tier is to specify the sequence of steps needed to complete each procedure in Tier 3. In addition, the applicable product and process standards are listed. A separate procedure step tier exists for each phase.

The purpose of each tier is to progressively elaborate the activities within each phase. The tiers relate to each other in specific and well documented ways. Sets of tiers, such as tier 1, tiers 1 and 2, or tiers 1, 2 and 3, form a self-contained graphical representations of process phases.

The traceability constraints insure that each construct shown in every tier has a well-documented, consistent relationship with associated constructs on the next higher tier of abstraction as well as the next lower tier of detail. A structured numbering scheme permits a task to be traced to the procedures implementing the task. Similarly, procedure steps are be unambiguously accumulated into the procedure the steps implement. Verification of the process representation from tier to tier is supported through additional traceability constraints and generation rules.

A complete listing of the model generation rules is contained in [HENJ92a]. The purpose of generation rules is to provide the resulting models with cohesive tiers. The rules are governed by the principles of information hiding, functional decomposition, and stepwise refinement. In addition, the rules encourage progressive elaboration through the more detailed tiers.

The modeling technique successfully integrates organizational, functional and behavioral information into a multi-layered process model. The multiple layers provide visibility into the software process used by an organization, as advocated by the SEI. The addition of measurements to the model depicts what type of data is extracted and where. The data is used in statistical process management as required by TQM.

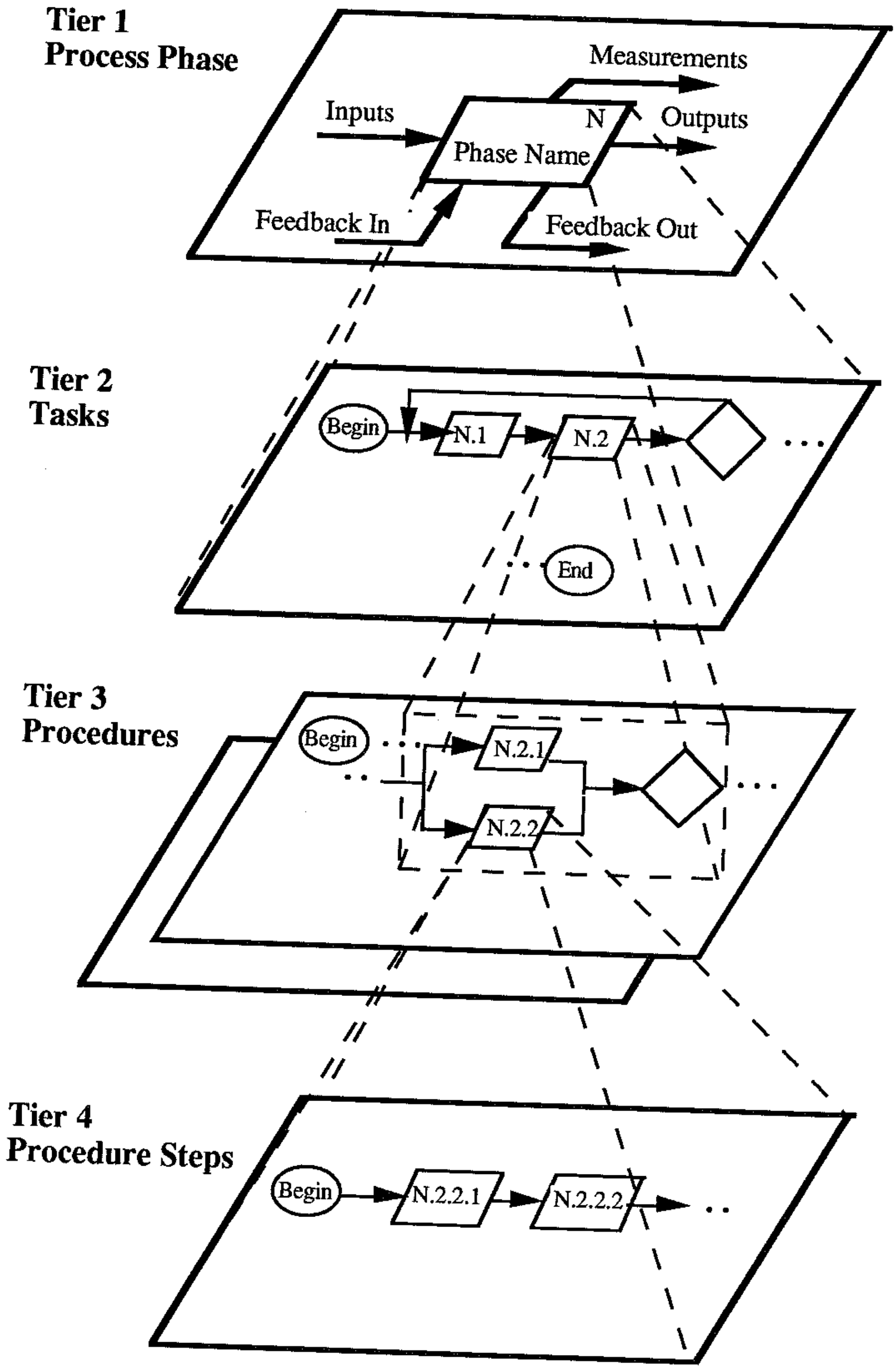

Figure 2. Process Modeling Tiers

3.3 The Data Gathering Techniques

The data gathering techniques described in this proposal specify how a software process is instrumented. The data gathering techniques:

- classify the data such that it is amenable to statistical analysis,
- define the data clearly and accurately, and
- require review of data definitions.

What data to collect and where in the process to collect the data is dependent on investigation results, the process model and organizational priorities. Process investigation may discover high-priority problem areas within the process which need immediate improvement. The construction of a process model may suggest easily obtainable data about a specific problem within the process. An organization may already recognize a significant, recurring problem within the process needing improvement.

The purpose of this step in the assessment methodology is not to propose a new set of process and product measures but rather to advocate that data gathering be performed. The data is selected by using investigation results, the process model, organizational priorities and previous software measurement work. The data is specified using a defined procedure and a data definition form.

Selection of data to collect should not be performed without a specific purpose and guidelines. The data to be collected must be carefully defined. Haphazard collection of a wealth of data is unlikely to be useful. The assessment methodology requires data to be collected from two different approaches using a data definition form.

Two specific approaches to data collection are used: horizontal and vertical collection. These approaches are shown in Figure 3 and defined as follows:

1. Horizontal - Data about a single activity or product is collected over time or successive process phases to determine characteristics or trends. For example, the number of errors found during successive test phases of a single specification change would be collected.

2. Vertical - Data about a group of activities or products is collected at a single point in time. For example, the estimated number of man-months needed to implement all specification changes approved during Month 2 in Figure 3 would be collected.

These two approaches organize the data acquired in several important ways. Vertical data can be viewed as a snapshot of project data at a particular point in time, making the data suitable for use as status and project review data. Horizontal data can be viewed as process and project tracking data, which is useful for discovering trends in the project or process. Classifying data as vertical and horizontal allows the application of statistical analysis techniques to the data. These techniques include analysis of variance, multiple regression, analysis of covariance and categorical independence techniques.

The proposed assessment methodology advocates collection of data based on the work of Boehm [BOEB81], Mogilensky [MOGJ90], SEI [SEI91], Weiss [WEID85] and Huff [HUFK86]. Post-delivery data based on the work of the Rome Air Development Center [CHRA87a], [CHRA87b], Schneidewind [SCHN87], and Rosson [ROSC87], are also suggested. Prospective data should be tailored to the organization and the process used. Both data definition and tailoring are more easily performed when data descriptions use defined procedure.

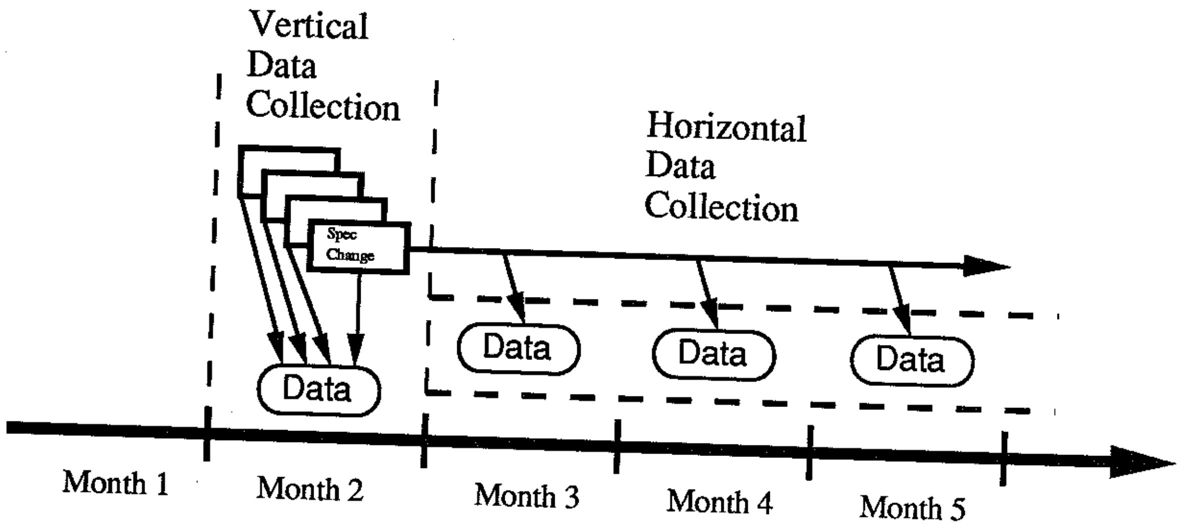

Figure 3. Vertical and Horizontal Data Collection

The procedure advocated in this assessment method is as follows:

1. Describe what characteristic the data measures.
2. Define why the characteristic is of interest.
3. Describe how the data will be obtained.
4. Specify what the form of the data (units, frequency, etc.), and
5. Specify how the data will be validated.

The following form defines the collected data :

Data Definition Form

Characteristic:	This section describes the characteristic of interest.
Justification:	This section describes exactly why the data is collected.
Measurement Approach:	This section describes how and when the data is collected.
Data:	This section specifies the form (units, frequency, formula) of the data.
Validation:	This section describes how the data item is validated.

An organization's confidence in the validity of the data, and support in the collection of the data, is crucial to obtaining meaningful data. The data definition form specifies the data to be collected in a logical, understandable form. Data definition forms are reviewed by the personnel involved in the software process. Data definition review involves the organization and is critical to gaining the support of software process personnel.

In order to move Software Engineering toward recognition as a true engineering field formal methods must be applied. Scientific investigation utilizes a structured approach to gathering data about an object or process. A structured approach insures the data gathered accurately reflects the characteristics of interest. The vertical and horizontal data-collection approaches structure data gathering. The data definition form and supporting review insures that the data is clearly specified and completely reviewed. Data gathered using the techniques described here is amenable to many types of statistical analysis methods.

3.4 The Analysis Techniques

The goals of the analysis step are to consider both the necessary and sufficient conditions in the assessment. Consideration of the necessary conditions is based on process investigation and the process model. Consideration of the sufficient conditions is based on the process and post-delivery data. Analysis techniques for these two different types of considerations are presented separately.

3.4.1 Process Content Analysis

The information obtained during investigation includes the activities comprising the process and the environmental factors affecting the process. The process model documents the activities used and the structure of the activities within the process. This information is used to answer the following significant questions:

- What activities are missing from an organizations process?
- What activities should be added to the organizations process?
- Where in the process should these activities be added?

Content and structure of the process is compared with the detailed and extensive Capability Maturity Model for Software(CMM) produced by the SEI [SEI91]. The CMM describes the activities appropriate for process maturity levels 2 though 5. The activities are classified into major categories such as Project Management, Process Management, Software Quality Assurance, Software Configuration Management, etc. While the content and interpretation of the CMM continues to evolve, it does represent a significant repository of accepted software engineering principles and project management practices. The purpose of this comparison is not to force an organization to adhere strictly to the process defined in the CMM, but rather to discover missing activities beneficial to the organization. Previous experience in this area has shown the major benefit of this type of analysis is in the utilization of accepted principles and practices by an organization [HENJ91a] [HENJ91b].

Environmental factors are a significant considerations in the assessment of an organization. These factors often explain why process activities exist, don't exist, or why a process works in a particular fashion. Previous assessment of small organizations discovered that very different organizational goals significantly impact the software process used [HENJ91a]. For example, the emphasis on error prevention is very different if an organization produces proof of concept software rather than performing long-term maintenance on an established product. Careful consideration of environmental factors allows the appropriate software engineering principles and practices to be selected for inclusion in a software process.

3.4.2 Statistical Analysis of Process Output

Software project managers and division managers typically lack quantitative data regarding project status, progress, and product quality. Given this lack of objective data, the affect of man-power loss, specification changes, error-rates and other process factors are estimated using rules of thumb, past experiences or "gut feel." Such estimates leave a large margin for error. More mature engineering disciplines rely on quantitative data for determining project status, progress, and product quality. Software Engineering is in need of quantitative project management based on objective data and statistically valid relationships between and among the data.

Two specific types of relationships are of interest. First, relationships among process data are examined to determine where statistically valid prediction can be performed and where significant differences in process data values exist. Secondly, relationships between process data and post-delivery product data are examined to determine if relationships exist, what type of relationship exists and what process data is statistically valuable in predicting the post-delivery product data.

Some of the questions which are empirically evaluated are:

- Can future process and product characteristics be predicted from process data?
- What relationship exists between process data and post-delivery product data?
- What is the relationship between projected data and actual data?
- What is the cumulative affect of horizontal data on the post-delivery product?
- What is the combined affect of process activities on the post-delivery product?

Data acquired during the software process and following delivery of the product are subjected to statistical analysis. Analysis of process output employs accepted statistical methods, such as multiple regression, analysis of variance and covariance, and analysis of categorical data. Multiple regression is employed to determine what type of relationship, if any, exists among process data and between process data and post-delivery data. Data is combined by time period or logical grouping, then compared using analysis of variance techniques to detect significant differences.

Statistical analysis satisfies the need for qualitative process evaluation and is the basis for fact based decision making. The effectiveness activities present in the process is determined using statistical analysis. In addition, the impact of process activities can be shown using statistical techniques. Fact based decision making is based on valid statistical predictions. The quantitative impact of process activities and trends on the delivered product are also used in making fact based decision.

5 Implementation

Implementation of the integrated assessment methodology is underway within the Computer Program Projects Group of General Electric Corporation. The Computer Program Projects Group is responsible for the management of software projects, primarily the computer programs used by the AEGIS Radar and Weapon Systems. AEGIS is the longest running Naval subcontract existing today, employing thousands of people and costing billions of dollars.

The assessment methodology described in this proposal involves four significant steps, each of which is a large task in itself. Implementation of the integrated assessment methodology and formulation of resulting improvements are the tasks of the Product Improvement Committee(PIC) [GESD91].

The investigation step is complete. Dozens of interviews were conducted involving such diverse groups as quality assurance, configuration management, program management, subcontractors, testing, Naval auditing agency. The inputs of all these groups was critical because each group has a unique perspective, and distinct and sometimes conflicting goals. An estimated SEI evaluation was also performed using a technique developed by Henry et al.[HENJ91a].

The modeling step is complete. The models produced underwent extensive review throughout the AEGIS development community. Complete documentation of the development process is based on process models constructed using the modeling technique described in section 3.2. The graphic models are supplemented with task entrance and exit criteria, and specification of the applicable General Electric computer program standards, work instructions and directives.

The data collection step is ongoing [GESD91]. Several versions of the AEGIS Radar System, referred to as baselines, are developed concurrently. Process data has been collected on different baselines at different points within the process. The data is collected by the various groups within the AEGIS development community and submitted to the PIC. This data is stored in a tightly controlled metrics database.

Process content analysis and statistical data analysis.are also ongoing. Initial process content analysis has been completed, and is detailed in "Improving the AEGIS Computer Program Development Process to Achieve SEI Maturity Level 3" [HENJ91b]. Statistical analysis of requirements volatility and errors/defects is underway. Initial results are encouraging, for example, a regression equation predicting the effort required to implement each upgrade to a baseline of the AEGIS computer program has been developed. The R-square coefficient of this equation is .92. More detailed results on statistical analysis are forthcoming [HENJ92b].

The assessment methodology continues to evolve. Process documentation is refined, and reflected in the modeling technique where appropriate. Analysis of both process content and data indicates additional data needed and modifies existing data definition. Statistical analysis highlights activities within the process where the greatest leverage can be gained on process and product improvements.

6 Conclusions

As competition for software contracts increases, the need for a highly visible, quantitatively evaluated software process will become critical. Software organizations continually improving their software processes will enjoy a significant advantage in acquiring contracts based on a their history of producing quality software on time and within budget.

The existence of a structured, validated assessment methodology is the foundation for process improvement. In fact, it can be argued that process improvement is only as effective as the assessment it is based on and evaluated by [HUMW89]. The assessment methodology described here is a very effective approach to continuous process assessment and improvement.

An approach to process assessment integrating TQM and SEI approaches provides insight into both the content and the effectiveness of an organization's software process. Process content improvement places effective process activities within the software process. Statistically valid relationships among and between the data evaluates the effectiveness of an organizations process. The results can be used in project management, process management, product quality evaluation and risk assessment. The use of quantitative data in each of these areas will move the field of Software Engineering toward a more advanced stage of technical development.

References

[ARTJ86] Arthur, J.D., Nance, R.E., and Henry, S.M., *A Procedural Approach to Evaluating Software Development Methodologies: The Foundation*, Technical Report SRC 86-008, Systems Research Center and Department of Computer Science Virginia Tech, 1986.

[BOEB81] Boehm, B. W., *Software Engineering Economics*, Prentice Hall, 1981.

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

[BOYJ91] Boyle, J., Executive Vice President, Video Lottery Consultants Inc., Bozeman Montana, personal communication, December 7, 1991.

[CHRA87a] Chruscicki, A. J., "Software Reliability Assessment," Rome Air Development Center, Griffiss AFB, 1987.

[CHRA87b] Chruscicki, A. J., "Software Maintainability Prediction and Assessment," Rome Air Development Center, Griffiss AFB, 1987.

[CRAS85] Crawford, S.G.and Fallah, M.H., "Software Development Process Audits - A General Procedure," *Proc. of the 8th International Conf. on Software Engineering*, August 1985.

[DOD88] "Total Quality Management Master Plan," Department of Defense, August 1988.

[GESD91] "AEGIS Computer Program Metrics Plan," General Electric Corporation, Government Electronic System Division, March 1991.

[GRAR82] Grady, R. B. and Caswell, D. L., *Software Metrics: Establishing a Company-Wide Program*, Prentice Hall, 1987.

[HENJ91a] Henry, J. E., Keenan, S., Keenan, M. and Henry, S., "An Assessment Method for Small Organizations," *Proceedings of the 1st International Software Quality Assurance Conference*, October 1991.

[HENJ91b] Henry, J. E. and Dublanica, W., "Improving the AEGIS Computer Program Development Process to Achieve SEI Maturity Level 3," General Electric Corporation, Government Electronic System Division, November 1991.

[HENJ92a] Henry, J. E., Dublanica, W., Kubeck, J. and McMullen, B., "A Process Modeling Technique and Application Results," *Proceedings of the Software Tools and Techniques Symposium*, March, 1992.

[HENJ92b] Henry, J. E., Henry, S., and Matheson, L., "Quantitative Maintenance Management," submitted for publication, *Journal of Software Maintenance*, April, 1992.

[HUFK86] Huff, K. E., Sroka, J. V. and Struble, D.D., "Quantitative models for managing software development processes," *Software Engineering Journal*, January 1986.

[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., "Software Process Improvement at Hughes," *IEEE Software*, July 1991.

[JABJ91] Jablonski, J., R., *Implementing Total Quality Management: An Overview*, Pfeiffer, 1991.

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

[OSTL87] Osterwell, S., "Software Processes are Software too," *Proceedings of the 9th International Conference on Software Engineering*, 1987.

[QUIR80] Quinnan, R.E., " A programming process study," *IBM Systems Journal*, Vol. 19, No. 4, April 1980.

[ROSC87] Rosson, C. V., "Management Indicators: Assessing Product Reliability and Maintainability, Technical Reports SRC-88-011, Systems Research Center and Department of Computer Science Virginia Tech, 1988.

[SCHN87] Schneidewind, N. F., "The State of Software Maintenance," *IEEE Transactions on Software Engineering*, Vol. 13, No. 3, March 1987.

[SEI91] "Capability Maturity Model for Software," CMU/SEI-91-TR-24, Software Engineering Institute, Carnegie Mellon University, August 1991.

[SEIS92] Siegal, S., "Why We Need Checks and Balances to Assure Quality," *IEEE Software*, January 1992.

[WEID85] Weiss, D.M. and Basili, V.R., "Evaluation of Software Development by Analysis of Changes: Some Data from the Software Engineering Laboratory," *IEEE Transactions on Software Engineering*, Vol. 11, No. 2, February 1985.