# Applying Software Engineering Principles to Process Modeling

*Joel Henry*

**TR 92-26**

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

May 18, 1992

# Applying Software Engineering Principles to Process Modeling

Joel Henry

## 1. Introduction

The production of high quality software has become one of the most important goals in computing. The software development process has a significant impact on the quality of the developed product[ARTJ87]. In response to this recognition, the software development process has received much attention throughout industry[HUMW91], academia[HUMW89] and government[DOD85]. The focus of much of this attention is on understanding, assessing and improving the software process and the software products produced.

Software products, including both executable code and supporting documentation, are produced by the implementation of a software process. The software process consists of the activities, tools and methods used to produce software products[HUMW89]. In order to understand, assess and improve a software process, the process must be accurately documented. An important part of process documentation is a graphic model of the development process.

Process modeling has become an active research area in the recent past, as demonstrated by the abundance of manuscripts published in workshops and conferences during the last five years(i.e. The International Conference on Software Engineering, The International Software Process Workshop and The Hawaiian International Conference on System Sciences). Significant effort has been devoted to process modeling at the Software Engineering Institute[KELM89]. Process modeling is also a critical component in the application of Total Quality Management to software development[BLAR92].

Process models document the development process by:

- communicating the roles and responsibilities of both individuals and groups
- specifying when software products are created and changed
- depicting where measurement is performed
- supporting process improvement
- describing the communication channels between development groups

Process models are constructed using specific modeling methods or techniques. These techniques impart certain characteristics to the models they produce. Development of a modeling technique should begin by specifying the characteristics desired in the resulting models.

The characteristics desired in our models include:

- clear representation of the flow of control between and among process activities
- cohesive integration of functional, behavioral and organizational views within a single model
- representation of the process in multiple levels of implementation detail
- simple integration of process measurement
- careful documentation of the relationship between product and process

Imparting these characteristics to process models requires a modeling technique employing well-structured construction principles. Application of the software engineering principles of information hiding, top-down functional decomposition and stepwise refinement to process modeling imparts many desirable attributes, including modularity, strong cohesion and weak coupling, to the models produced. In addition, allowing representation of the model in multiple levels of detail makes the models usable by a wide variety of project personnel, from coders to upper level managers.

## 2. Background

The development of a new modeling technique arose from the need to produce a standard method for representing the phases within the software process at a large-scale military software development organization. Previous attempts at phase modeling produced models with widely varying levels of detail and very different modeling constructs. Confusion resulted within and among the groups responsible for implementing process phases. A standard technique for representing activities within each phase was needed. In addition, an ongoing effort to implement software metrics required accurate phase models to insure a well-understood, well-documented software process.

The software process consists largely of activities and the transfer of control between activities. Control flow diagrams represent these types of relationships very well and are the basis for the modeling technique.

Control flow diagrams alone do not sufficiently describe the software development process. Design methodologies using control flow diagrams are often supplemented with several types of supporting documentation. The supporting documentation includes timing specifications, data flow diagrams, and state transition charts. Process modeling requires the representation of similar, although not identical, information.

While other modeling techniques advocate the use of multiple models representing behavioral, organizational and functional views[KELM89], multiple process models based on dissimilar views of a development process are confusing to development personnel. Rather than produce multiple models of a software process based on different views, additional constructs represent product flow, schedule constraints and other external influences.

The process models are used by a military contractor, and therefore must meet the applicable military standards. In light of the development environment, the organization specified the resulting process models be:

- auditable by technical representatives of the customer,
- descriptive of the process in multiple levels of detail,
- accurate descriptions of activities within the process,
- maintainable over a period of years,
- defined by significant process phases, and
- understandable by the wide variety of development personnel(including configuration management, quality assurance, project management, developers, and subcontractors).

## 3. Overview of the Modeling Technique

Model construction must be carefully constrained to produce phase models with the characteristics desired. 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, be used to construct process models. Second, the purpose of each tier was defined. Third, model traceability constraints were specified to insure consistent representation of a process between different tiers. Finally, rules governing the generation of models through all tiers, and specific rules for the generation of each tier were defined.

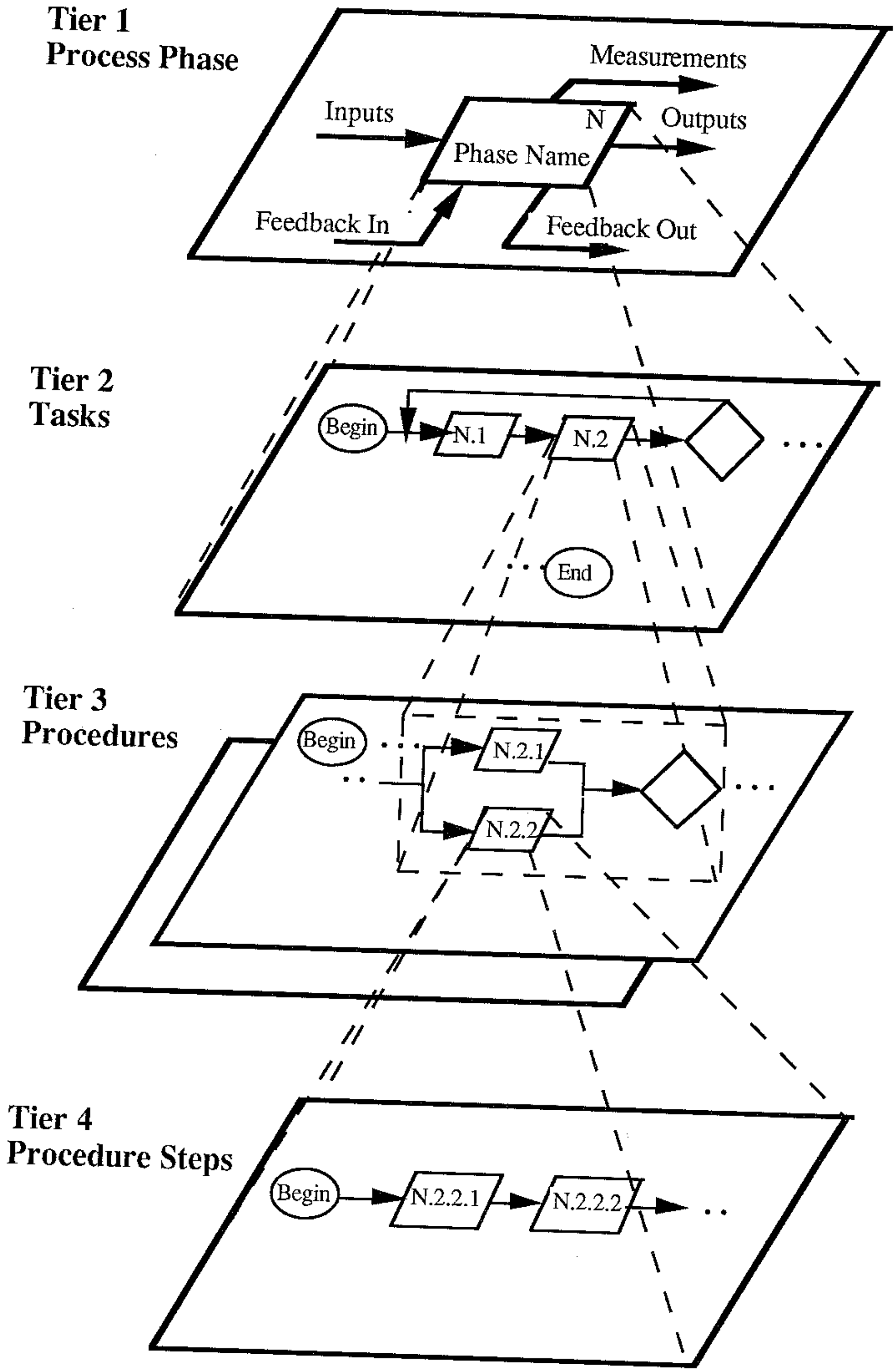


**Figure 1. Modeling Tiers**

The four tiers used in this modeling technique are shown in Figure 1 and named and described as follows:

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

2. Tier 2 - Task Tier. The purpose of this tier is to describe <u>what</u> major tasks comprise a phase. A separate task tier exists for each phase.

3. Tier 3 - Procedure Tier. The purpose of this tier is to specify <u>how</u> the major tasks of Tier 2 are performed. Tier 3 describes the major implementation steps for each task shown in Tier 2. 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, Tier 3 lists the applicable product and process standards. 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 self-contained graphical representations of process phases.

The traceability constraints insure each construct shown in tier a phase has a well-documented relationship to constructs on the next higher tier of abstraction as well as the next lower tier of detail. A structured numbering scheme insures each task consists of a single set of procedures. Similarly, procedure steps unambiguously implement a single procedure. Additional traceability constraints and generation rules support tier to tier verification of the process representation.

Section 4 presents a complete listing of the model generation rules. The purpose of generation rules is to provide the resulting models with cohesive tiers. The rules implement the principles of information hiding, functional decomposition, and stepwise refinement. In addition, the rules encourage progressive elaboration through the more detailed tiers.

## 4. Model Generation Rules

The modeling technique contains specific definitions, symbology, generation rules applicable to all tiers and generation rules governing each specific tier.

A standard set of terms defines the information added to control flow diagrams. The terms and their definitions are as follows:

<u>Inputs</u> — Products delivered at the start of a phase from a previous phase or from external sources. Inputs initiate the current phase.

<u>Outputs</u> — Products generated by a phase that are inputs to a subsequent phase or are necessary documentation at Tier 1.

<u>Feedback In</u> — A change, constraint or other impact accepted by a phase generated by another phase at Tier 1.

<u>Feedback Out</u> — A change, constraint or other impact generated a phase accepted by another phase at Tier 1.

Measurements — Values obtained during a phase used to quantitatively evaluate the process, product or phase.

The symbology used in this modeling technique is shown in Figure 2.

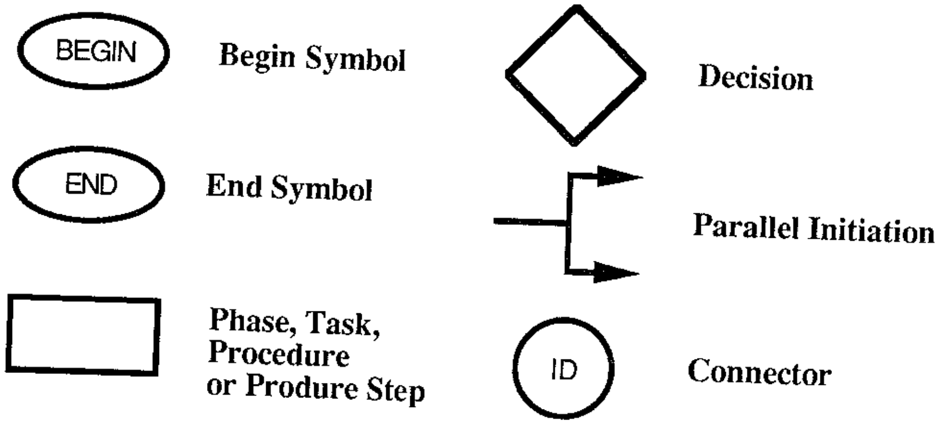


**Figure 2. Modeling Symbols**

Eight generation rules govern construction of models throughout all tiers. These rules are:

1. All Outputs and Feedback Out shown on a tier 1 phase block shall have clearly identified generation blocks on tiers 2 through 4 of that phase.

2. All Inputs and Feedback In shown on a tier 1 phase block shall have clearly identified blocks accepting the Input or Feedback In on tiers 2 through 4 of that phase.

3. A block on any tier shall have exactly one control flow entrance point and one control flow exit point.

4. A decision block at any tier shall have exactly one control flow entrance point.

5. Blocks at all tiers shall be uniquely numbered. Tier 1 phases are numbered sequentially starting with 01. Blocks on Tiers 2 through 4 are numbered by adding a period and a number to the parent number of the block they describe. The following is an example of the numbering sequence for related blocks on Tiers 1 through 4.

   Tier 1: 01 - phase 01
   Tier 2: 01.1, 01.2, 01.3, 01.4 - the four tasks of phase 01
   Tier 3: 01.2.1, 01.2.2, 01.2.3 - the three procedures task 01.2
   Tier 4: 01.2.3.1, 01.2.3.2, 01.2.3.3, 01.2.3.4 - the four steps of procedure 01.2.3

6. The single group within an organization responsible for a phase shall be stated in Tier 1. When the group performing a task, procedure, or procedure step is different from the group named in Tier 1, the description of the task, procedure or procedure step shall identify the responsible group.

7. Lines exiting a decision must go to a block or a different decision block.

8. All tiers shall start with a BEGIN symbol and terminate with and END symbol.

In addition to the general rules given above, specific generation rules for creating Tier 1 of a phase model are:

1. Tier 1 shall clearly depict each phase of the development process.
2. Tier 1 shall include for each phase the
   - Phase name and number,
   - Inputs,
   - Outputs,
   - Feedback In,
   - Feedback Out, and the
   - Responsible Group.
3. The phase is initiated upon receipt of the inputs.

The model generation rules for Tier 2 are:

1. Tier 2 shall provide an overview of the phase and should consist of four to ten tasks.
2. Each task shall have a single purpose and this purpose shall not overlap with any other task.
3. Tier 2 is initiated upon receipt of the inputs shown in Tier 1.

Specific generation rules for Tier 3 are:

1. Tier 3 shall be a depiction of the procedures needed to implement the tasks in Tier 2 and maintains the same control flow shown in Tier 2.
2. Each procedure block shall map to a single task shown on Tier 2.
3. The single control flow entrance point and exit point of each task shown in Tier 2 shall be maintained in Tier 3.

Tier 4 phase models shall conform to the following rules:

1. Tier 4 shall depict the steps needed to implement each procedure shown on Tier 3.
2. A separate representation shall be created for each procedure.
3. Associated with each step shall be the applicable process and product standards.

## 5. Example Model

Tiers 1 and 2 of the Engineering Test and Evaluation(ET&E) phase constructed using the modeling technique are presented in Figures 3 and 4. The ET&E phase is the final test phase before system qualification testing. A simulated deployment environment tests functional upgrades and specification changes. Proprietary concerns prevent the presentation of Tier 3 here. Specific Input, Output, Measurements, Feedback In and Feedback Out are similarly omitted.

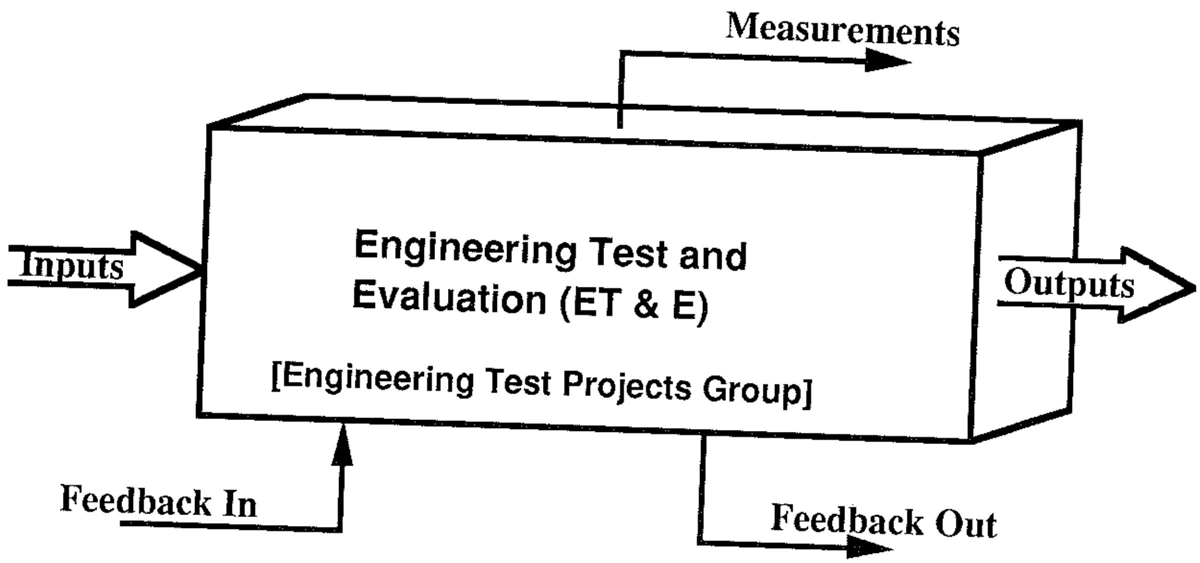


Figure 3. ET&E Tier 1

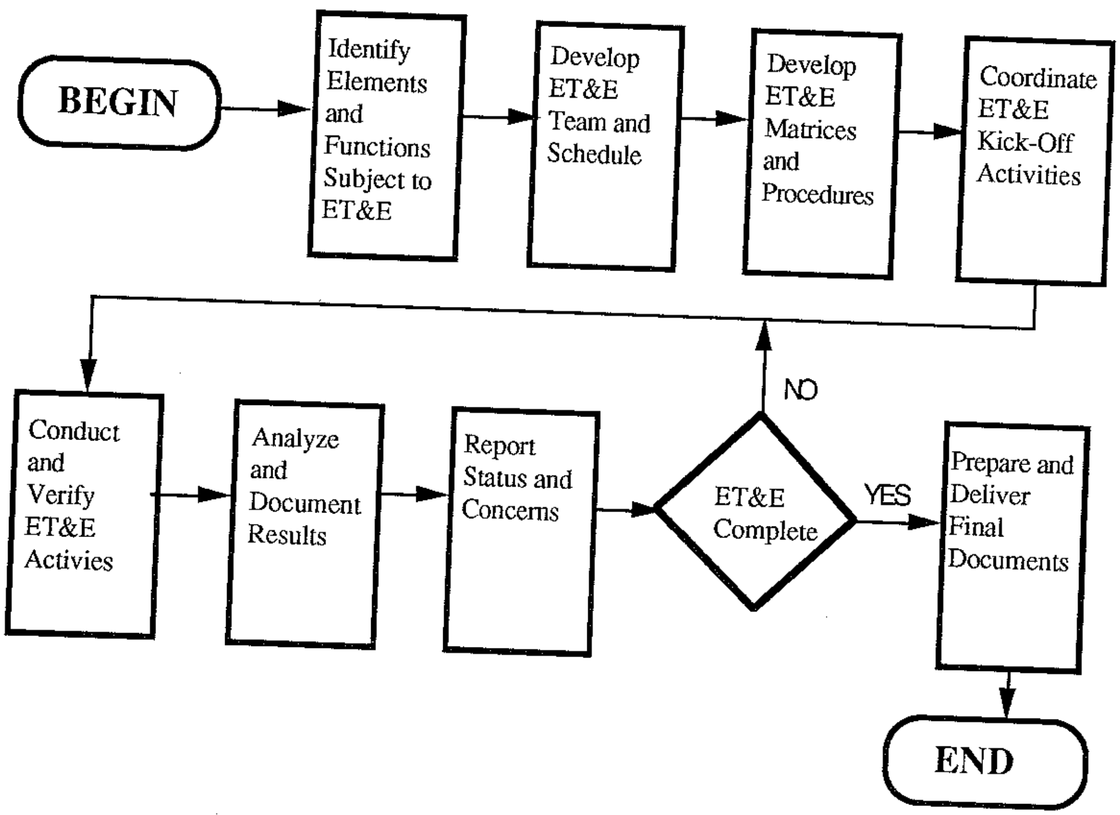


Figure 4. ET&E Tier 2

## 6. Evaluation of Modeling Technique

A small number of personnel, in conjunction with each group responsible for a process phase use the modeling technique. After initial training and practice with the technique, model builders found the technique to be easy to understand and apply. The model builders had little or no background in the area of model construction. Many of the initial difficulties encountered were the result of inexperience with process modeling rather than any particular problem with the modeling technique. Once the idea of applying of stepwise refinement, information hiding and functional decomposition to model building became more familiar, the model builders became proficient at creating and altering process models.

A tendency to prematurely specify details was observed. This is similar to the impulse of programmers to rush toward coding. Creating and reviewing each tier of a phase with the responsible group before creating the next more detailed tier corrected this tendency.

The organization determined initial phase models be specified only to Tier 3. Given the extensive use and required auditability of process activities, such a decision was appropriate. Additional detail will be added in the future based on experience with the models and greater knowledge of the development process.

While the modeling technique supports concurrent activities, some difficulty arose keeping the functional boundaries in Tier 3 consistent with those in Tier 2 when large amounts of concurrency existed. Some modelers felt the model generation rules should be relaxed in these cases, while others thought the difficulty resulted from an excessively complex, uncoordinated process phase. All agreed the existence of inflexible phase schedules required some concurrency. Discussion of this topic continues.

## 7. Evaluation of Models Produced

Process models are part of a larger effort to define the activities used by the organization. The models are supplemented by additional documentation defining the procedures used by and roles of software quality assurance, independent verification and validation, configuration management and subcontractors. Applicability and use of process standards, product standards, directives, work instructions and best practices also supplement the process models.

The organization tasked a committee to define each phase within the development process, from system requirements through acceptance testing. Recognizing the support and involvement of the responsible groups is critical to the accuracy and acceptance of the resulting phase models. The committee actively involves each group in the creation and refinement of their phase model. Group involvement and evaluation of the phase models constructed was very positive.

Reviews of the phase models are very positive. The models are highly cohesive within each tier. Communicative power is very good because of the effective use of information hiding and stepwise refinement. The view of a phase model is at an appropriate level of detail, depending on the audience and the intended use. For example, high-level managers use Tier 1, configuration management uses Tiers 1 and 2, and project managers use Tiers 1, 2 and 3.

Feedback In and Feedback Out are the communication channels between phases. These channels are phase interfaces and the information exchanged an indication of phase coupling. Accurate documentation of these channels controls coupling and supports efforts to reduce both the amount, and type of phase coupling.

Some difficulty arose regarding hard copy representation of Tier 3 phase models. For example, too many procedures per page become confusing while too few inhibits analysis of the process within a phase. Work continues in this area.

The phase models are maintained on a personal computer using a commercial data flow software package. Generation and maintenance of phase models are time-consuming but acceptable. The author felt very strongly that a modeling technique be developed independent of existing tools. A modeling technique based on an existing tool faces the limitations of the tool and typically fails to exhibit the characteristics desired in the resulting models.

The phase models constructed are to be in used for several years. Improvements to both the technique and the resulting models will continue. Additional process and product measurements will be integrated into the model supporting the application of Total Quality Management to software development.

## 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.

[BLAR92] Blasewitz, R.S, and Henry, J.E, *Total Quality Management Applied to Software*, To be published.

[DOD85] "DOD-STD-2167A, A Defense System Software Development," United States Department of Defense, June 4, 1985.

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

[HUMW91] Humphrey, W.S., Snyder, T. R. and Willis, R. R., "Software Process Improvement at Hughes Aircraft," *IEEE Software*, July 1991.

[KELM89] Kellner, M.I., "Software Process Modeling: Value and Experience," *SEI Technical Review*, Software Engineering Institute, Carnegie Mellon University, 1989.