Empirical Software Engineering, 3, 159–177 (1998)


# A Guttman Scaling of CMM Level 2 Practices: Investigating the Implementation Sequences Underlying Software Engineering Maturity

JOHN G. BILOTTA
*Charles Schwab & Co., Inc., San Francisco, CA 94104, USA*

JOHN F. MCGREW*
*SBC Services, Inc., Pleasanton, CA 94588, USA*

*Received June 19, 1997; Revised January 29, 1998*

**Abstract.** The Capability Maturity Model specifies several hundred key practices that must be implemented as a team moves from Level 1 to Level 5 of the model. However, the model does not specify within each level the optimal sequence in which to implement the practices. Level 2 contains 121 such practices grouped under six key process areas (KPAs) which are further subdivided into five common feature areas (CFAs). Although the KPA/CFA structure has a logical fit with the organizational structure of very large software development teams, it does not correspond to the reality of small teams. Using Level 2 audit data collected on 10 small software development teams, the authors try to determine whether the data itself can point to a more appropriate implementation strategy for small teams. The data is analyzed using Guttman scaling techniques (scalogram analysis). The results indicate that there is a single underlying, orderable dimension at Level 2 which lays out a step-by-step path upward from Level 1. The order of the items is found to map well to the familiar Plan-Do-Check-Act cycle widely used by project managers to organize and control work efforts. The extracted scale can be used as an assessment tool to provide management with a quick snapshot of a team's current position relative to Level 2.

**Keywords:** Capability Maturity Model, Level 2, Guttman scaling, scalogram analysis, implementation strategies

## Introduction

The Software Engineering Institute's Capability Maturity Model (CMM) provides a stage theory of the development of software engineering competence. It specifies five stages, or levels, of maturity: Initial, Repeatable, Defined, Managed, and Optimizing. At each level, the CMM specifies a detailed list of practices that must be implemented if an organization is to evolve from chaotic to optimized software engineering processes. The relationship between process maturity and project success in terms of schedule, performance, and quality has been demonstrated in more than one study. For instance, in an analysis of 52 defense industry software projects, Lawlis, Flowe, and Thordahl (1995) observed improved cost and schedule performance as process maturity increased. Brodman and Johnson (1996) found similar results in the form of increased productivity, quality, and schedule time.

* John G. Bilotta is currently at Charles Schwab and Co., Inc., 101 Montgomery Street, San Francisco, California 94104. John F. McGrew is currently at SBC Services, Inc., 4698 Willow Road, Pleasanton, California 94588. This paper describes work that was done while both authors were at Pacific Bell, a subsidiary of SBC Communications, Inc.

What the CMM does not specify is the underlying mechanism that controls the order of acquisition of key practices within each level. The optimal path by which a team may move from one level to the next is hidden somewhere among the hundred or more key practices that define the behavior at each level. In particular, the optimal implementation path from Level 1 to Level 2 (every team's first stumbling block) is unknown. The lack of a practical implementation strategy has several unfortunate consequences: teams tend to flounder as they debate where to start with Level 2; teams become paralyzed as they are overwhelmed by the sheer number of things to be done; and teams may attempt to implement practices before the groundwork of prerequisite practices has been laid. This results in "teflon" teams—practices simply don't stick to them. Data reported by Johnson and Brodman (1996) showed that, while the average team achieved Level 2 in 23 months, some organizations did not reach Level 2 even after 50 months.

It is not known whether Level 2 practices are implemented randomly. If they are not, then there are at least two possible explanations: the order in which items are implemented is determined by their ease of implementation; or, alternatively, they may be implemented in order by the cumulative support they provide each other. Johnson and Brodman (1996) lend support to this interpretation. Their research showed that "the project planning and project tracking and oversight KPAs had the greatest influence on schedule, and that the peer review KPA contributed the most to quality" (Johnson and Brodman, 1996, p. 100). It is not the ease of implementation that made these important but their contribution to project success that differentiated these KPAs from the others.

Taking an evolutionary viewpoint, the authors would argue in favor of the cumulative support interpretation. The idea is simply explained. To survive, each key practice requires a project environment that is minimally capable of sustaining it. Practices which are introduced into such "prepared" environments will survive because the prerequisite practices have been implemented. Unprepared practices will ultimately die out—they will not "stick." From this view then, implementation is not necessarily institutionalization. "Most likely, these organizations do not have the factors in place that contribute to an organization's ability to mature" (Johnson and Brodman, 1996).

The authors have been led to this interpretation by their observation of the project teams that they have worked with. The teams that will be discussed in this study were given CMM Level 2 audits after a two-year campaign by the organization to reach Level 2. We assume that what survived the implementation effort (and what was observed during the audit two years later) were those practices that were sustainable. What did not survive could not be sustained because the prerequisite tasks had not been implemented. Other authors (Brodman and Johnson, 1996; Johnson and Brodman, 1996) have been led to similar conclusions when trying to explain why some groups spend as long as fifty months without progressing to Level 2. Johnson and Brodman (1996) commented that it was most likely that "these organizations do not have the factors in place that contribute to an organization's ability to mature."

***Moving from One Stage to the Next***

To make the transition from Level 1 to Level 2, there are 121 key practices to be implemented. The Level 2 practices are grouped under six key process areas (KPAs): Requirements Management, Project Planning, Project Tracking and Oversight, Subcontract Management, Quality Assurance, and Configuration Management. Within each KPA, the practices are further subdivided among five common feature areas (CFAs): Commitment, Ability, Activities, Measurement, and Verification (Paulk, Curtis, Chrissis, and Weber, 1993). The same organizational principles are used at each level of the CMM from Level 2 through Level 5. Thus, the CMM forms a three-dimensional structure made up of five maturity levels four of which are subdivided into approximately half a dozen KPAs which are themselves subdivided into five CFAs. The hundreds of key practices required to make the transitions from level to level are distributed throughout the CMM structure based on their content. While efforts have been made to tailor the content of the CMM for improved usability by small teams (Johnson and Brodman, 1997), there has been no research on the optimal sequence for implementation of the Level 2 activities.

The process of transition from level to level by the acquisition of defined behaviors is the driving mechanism of all stage theories. With its stage theory base, the CMM provides a model of the maturation of software engineering skill within teams and organizations. The KPA/CFA structure of the CMM was designed to parallel the structure of large software development teams. As the appropriate practices are implemented at each level, a team's maturity, as judged by the model, increases. Maturational stage theories are common throughout the social and behavioral sciences. Piaget's theory of intellectual development is a familiar example (Piaget and Inhelder, 1969). In it he defines four stages in the development of intellectual capability from childhood to adulthood (sensori-motor, pre-operational, concrete operational, and formal operational). The skills acquired in later stages build upon the skills developed in earlier stages; and each stage is characterized by well-defined classes of behavior that the individual is capable of performing. The structure of the CMM follows the same pattern: skills acquired in later stages build upon skills developed in earlier stages with well-defined classes of behavior specified at each level. The behavioral set is cumulative.

By definition a stage theory implies three things: first, there is an order of acquisition to the stages or levels; second, attaining a new level requires attainment of all preceding levels; and third, lower-level performance is predictable from higher-level performance. It is possible to construct a measurement scale for behaviors that satisfy these ordering criteria making measurement, quantification, and diagnosis possible. Such scales are the basis of assessment used to measure achievement, aptitude, or attitude. Scaling techniques, which are widespread not only in psychology, sociology, and education, but also in political science and marketing, explore the dimensionality of behavioral data. They provide the tools to determine whether meaningful dimensions underlie behavioral models and the tools to identify the defining elements of those dimensions.

The CMM itself forms a five-point scale. Figure 1 illustrates this with an example using five hypothetical project teams as they might be rated on the scale defined by the CMM. Scales can be generated from theories (as is the case in Figure 1) or they can be extracted

| | CMM Level | | | | | Scale |
|---|---|---|---|---|---|---|
| Team | 1 | 2 | 3 | 4 | 5 | Score |
| A | + | + | + | + | + | 5 |
| B | + | + | + | + | - | 4 |
| C | + | + | + | - | - | 3 |
| D | + | + | - | - | - | 2 |
| E | + | - | - | - | - | 1 |

*Figure 1.* How a measurement or classification scale can be derived from the specifications of the CMM. A team which has satisfied all practices at all levels (indicated by a "+" at each level) is assigned a scale score of 5, or more commonly, they are described as a "Level 5 team."

from experiential data (as will be the case in this study). Note how coarse-grained the scale in Figure 1 is. Each point on the scale represents the successful implementation of a large number of individual practices. Patterns that do not match one of the five row patterns shown in the Figure 1 are considered impossible under the CMM, that is, a team or organization cannot succeed at Level 3 yet fail at Level 2. Note too that detailed information on team behavior within levels is not reflected in or extractable from this particular scale. The within-level activities are assumed by the model to be fully satisfied.

If we are looking for a high level CMM implementation strategy, then the scale in Figure 1 provides a straightforward one: to attain "the continuous optimization of software engineering processes" (Level 5), a team must first attain each of the previous levels in the order shown. At a detailed action-planning level, however, this is not the kind of value-added information a software project team needs.

To some extent, it is assumed by many practitioners that the KPAs and the CFAs themselves provide an organizing principle for the implementation of the key practices. Certainly, the KPAs and the CFAs have an intuitive, thematic validity. They correspond to the typical divisions of work within large project teams, but it is unclear whether they are valid for small project teams which do not have the resources to divide work responsibilities along the boundaries defined by the KPAs. Consequently, the appropriate implementation strategy within each level is not clear. The key practices do not have a clearly defined acquisition sequence as do the five levels themselves.

In their work with software project managers and practitioners, the authors have noted a tendency of many to view the CMM as a checklist of accomplishments, of work done rather than of ways to work. Our view of the CMM as a behavioral stage theory emphasizes that software engineering capability is about the process and sequence of doing, not just a loosely organized check list of things to do. It is focused on how teams work rather than the products they create in the same way that Piaget's theory of intellectual capability is focused on how people think rather than the facts they have learned (Piaget and Inhelder, 1969).

### *Research Questions*

The CMM can be viewed as a three dimensional structure: Level $\times$ KPA $\times$ CFA. The question of the number of dimensions that underlie the CMM and the question of the optimal path through the structure is of critical importance for implementation of the model. This is particularly true for small software development teams which lack the resources to use the KPA/CFA structure as an implementation guide. In this study, we ask whether there are one or more dimensions underlying the key practices at just one level—Level 2. If there are one or more dimensions, do they order the key practices in a meaningful way, that is, do they scale? Finally, does the ordering of key practices point to an implementation strategy more appropriate to small teams? In applying scaling techniques to behavioral data, we have both the tools and the raw material to understand what is being measured, to quantify how well it is being measured, and to refine the measurement instrument in the process.

One consequence of such an analysis is that we will be able to determine how well the underlying dimensionality of Level 2 is related to its a priori thematic structure of KPAs and CFAs. A second, and more practical consequence, is that we may be able to develop an optimal ordering of Level 2 practices. By ordering the key practices, we can identify a practical implementation strategy inherent in the structure of the scale itself. A third consequence is that we may be able to construct a diagnostic scale from a subset of the 121 items at Level 2, a scale that managers can use to gauge the development of their teams.

## Method

As part of an effort to assess the capability of software engineering teams within a large telecommunications firm, audits were performed on ten teams against Level 2 of the CMM. The teams were responsible for development of information system and network monitoring applications. The teams ranged in size from 5 to 16 members. The youngest team had been together for only 9 months and the oldest team had been in place for 9 years. The teams were audited by a three-person quality assurance group on the commitment, ability, activity, measurement, and verification items in the six key process areas at Level 2. This provided a total of 121 practices or items. Each item was scored as fully satisfied (2 points), partially satisfied (1 point), or not satisfied (0 points), yielding a maximum possible score of 242 points and a minimum score of 0. Observed scores ranged from 78 to 192 with a mean of 134.

The 121 items at Level 2 of the CMM were analyzed using Guttman scaling (Gorden, 1977; Shye, 1978). Guttman scaling (or scalogram analysis) falls under a class of analytical techniques known as non-metric scaling—so called because the techniques can be applied to dichotomous or ordinal data (Weller and Romney, 1990). Because non-metric techniques evolved in the behavioral sciences and may be less familiar to the software engineering community, we provide a brief explanation of scalogram analysis as used in this study.

| Subjects | Items a | b | c | d | e | f | g | h | I | j | k | l | m | n | o | p | q | r | s | t | u | Subject Score |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| A | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 20 |
| B | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 19 |
| C | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 18 |
| D | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 17 |
| E | 2 | 2 | 2 | 2 | 2 | 2 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 16 |
| R | 2 | 2 | 2 | 2 | 2 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 15 |
| G | 2 | 2 | 2 | 2 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 14 |
| H | 2 | 2 | 2 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 13 |
| I | 2 | 2 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 12 |
| J | 2 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 11 |
| Scale Score | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | |

*Figure 2.* A scalogram of a perfect Guttman scale. The goal of scalogram analysis is to simultaneously order both the subjects and the items from highest to lowest with no reversals of position. The scaling patterns shown here meet that criterion, given a sample of 10 subjects evaluated on 21 items where each item is scored 2, 1, or 0.

### *Scalogram Analysis*

Scalogram analysis was developed by Louis Guttman for use in social research to determine whether sets of dichotomous or ordinal data, such as the data collected in this study, contain one or more underlying scaleable dimensions (Guttman, 1944). A dimension is considered scaleable if it meets the following criteria. First, the subjects (in this study, the project teams) can be ranked from highest to lowest by their total scores (the row scores as shown in Figure 2) and, simultaneously, the items (in this study, the key practices) can be ranked from highest to lowest by their scores (the column scores as shown in Figure 2). Second, there should be no reversals of rank among either subjects or items. A data set, such as that in Figure 2, which has been ordered in this way is referred to as a scalogram. In a scalogram of a complete Guttman scale, no item is different from its immediate neighbors by more than one position.

Figure 1, referenced earlier, is also a scalogram of a complete Guttman scale. The items in Figure 1, representing the five levels of the CMM, are dichotomous. Either they are satisfied or they are not. The items in Figure 2 are ordinal, having three possible values (2, 1, and 0). It is easy to see that as the number of items increases (this study will analyze 121 items with three possible values each), the display of the scalogram can become very cumbersome. A more efficient way of displaying the results is illustrated in Figure 3. Figure 3, a scalogram chart, was generated by performing a principal components analysis on the Euclidean distances among the items in the scalogram in Figure 2. It provides a visual analogue of the intercorrelations among the items in the scale. Our use of this display format is consistent with formats used by others (Borg and Shye, 1995). The scalogram chart yields the same sequence of items as the manual Guttman scaling process but is compact and visually compelling.

The Guttman scale in Figure 2 has 21 possible scale positions providing a scoring range of

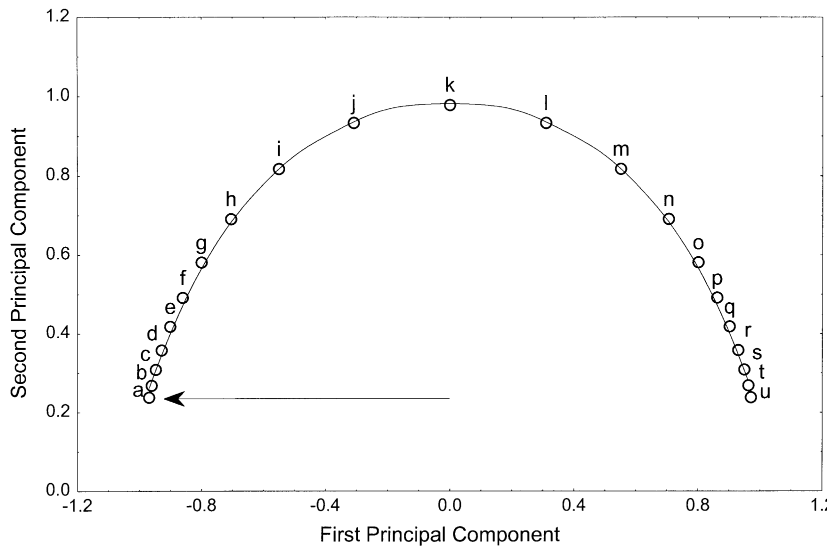


*Figure 3.* A scalogram chart generated by performing a principal components analysis on the Euclidean distances among the items in the scalogram shown in Figure 2. Rotate the arrow clockwise to read the scale in sequence from the highest scoring item (everyone satisfies this item) to the lowest scoring item (no one satisfies this item).

0 to 20. The column patterns in Figure 2 catalog the only permissible patterns for a sample with the same number of subjects (10 project teams) and the same range of item responses (0 to 2) as the data analyzed in this study. For that reason, the curve in Figure 3 can be used as a reference line representing the ideal distribution of items in a complete Guttman scale for a data set with the same characteristics as our sample. Note that scalogram charts are being used here only to simplify the presentation of results. Although visually useful, they do not provide a convenient method to determine scaling error in the items. For that reason, the scalogram analysis done in this study was performed using the standard manual method described in Gorden (1977).

### *Error in Scalogram Analysis*

Items with scoring patterns that violate the exact criteria of a Guttman scale are considered to contain scaling errors. For example, the item pattern

2 2 2 2 2 1 2 1 1 1 Ex. 1

has the same total item score (16) as item e in Figure 2. However, it violates the ordering property of a Guttman scale in the sixth position from the left where a 1 appears instead of the expected 2. Guttman's procedure would assess Example 1 one error—in the sixth position. Continuing in this manner, reviewing each item in turn, we can accumulate the total number of positional errors in the data set. The total observed positional errors can be used to generate a measure of overall scale consistency called the coefficient of reproducibility (*CR*)

$$CR = 1 - \frac{ObservedPositionalErrors}{MaximumPossibleErrors}$$

where “maximum possible errors” is simply the number of subjects times the number of items.

A pure Guttman scale has a *CR* of 100%—it contains no positional errors. The CR can range from 0% to 100%. The rule-of-thumb for an acceptable scale is 80% to 100% (Gorden, 1977). The *CR* represents the extent to which a subject’s item response pattern can be reproduced from his or her total score alone. The process of extracting a Guttman scale from our sample is the process of winnowing those items with the greatest error. In the end, we will achieve the most complete Guttman scale possible while minimizing the associated error, that is, while attaining the highest possible coefficient of reproducibility. Although there are alternative procedures for counting the errors in an item (McIver and Carmines, 1981), we use Guttman’s original procedure in this study.

It is not a foregone conclusion that a Guttman scale will be found in a data set. Item reduction by error minimization may converge on a Guttman scale but convergence is not guaranteed. Not all behavioral data contain dimensions that meet the criteria of a scale much less a Guttman scale. If it is possible to extract a Guttman scale, however, the analytical process has several important consequences. First, a scalogram analysis partitions the original item set into scaleable and non-scaleable classes based on the magnitude of scaling error in each item—thereby reducing the number of items required to measure the underlying phenomenon. Second, if there is a single underlying dimension, then a Guttman scaling analysis will find it and the items comprising the scale will form an ordered sequence of increasing complexity, ranging (in this study) from “most frequently satisfied” to “least frequently satisfied.” Referring again to Figure 2, item $a$, at the starting point of the scale, is the item most frequently satisfied. Item $u$, at the opposite end of the scale, is the item least frequently satisfied.

**Results**

A scalogram analysis of the 121 key practices at Level 2 of the CMM revealed the presence of a strong underlying unidimensional scale. The coefficient of reproducibility for the full set of 121 items was 79%, a surprising value considering that the key practices specified at Level 2 were placed there based on their conceptual validity and were not specifically chosen for their scaleability. Figure 4 presents a scalogram chart for the 121 Level 2 practices plotted against a curve representing a pure Guttman scale for a sample of the same size.

The scaling process can refine a scale by selecting for a subset of items with fewer scaling errors. A first pass analysis of the data selected those key practices with 2 or fewer positional errors, reducing the list to 75 items with a coefficient of reproducibility of 87%. A second pass selected from that subset those items with 1 or no errors, reducing the list of

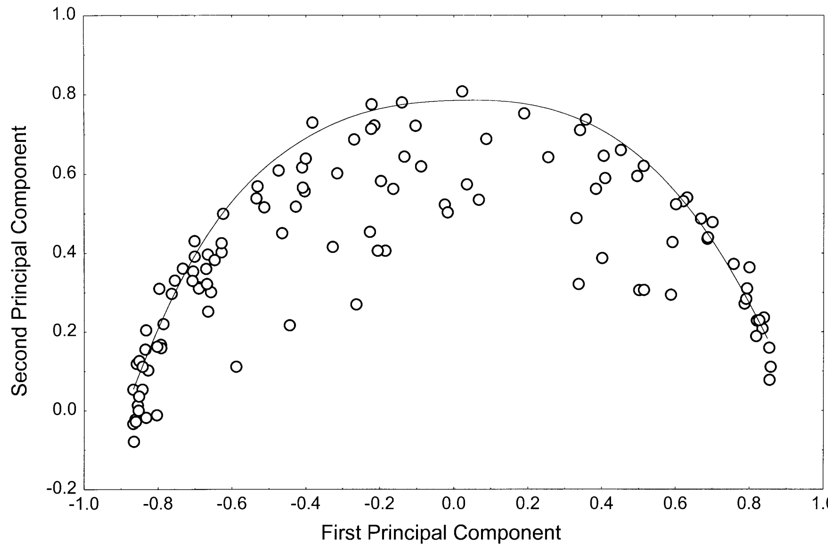


*Figure 4.* A scalogram chart of the 121 practices at Level 2 plotted against a pure Guttman scale (solid curve). When read clockwise from the left, items appear in sequence from most frequently satisfied to least frequently satisfied as evidenced by the 10 project teams in this study. Deviations from the pure scale line reflect the extent of scaling error in the items. The coefficient of reproducibility for the full item set is 79%.

scaleable items to 46 with a reproducibility of 93%. Efforts to reduce the error rate further were unsuccessful because further reduction would leave barely half the 21 scale positions needed to construct a complete Guttman scale for this sample. Figure 5 is a scalogram chart of the 46 scale items. The text of the 46 items appears in Table 1. In the table, each CMM item is identified by a code consisting of a two-letter abbreviation for the KPA, a two-letter abbreviation for the CFA, and a digit representing the item number. Thus, PT.AC.6 stands for Project Tracking & Oversight, Activity 6. A separate scalogram analysis was performed on all items discarded during the item reduction process to determine if a second underlying dimension could be found but none was identified. The search for a second dimension produced, at best, a coefficient of reproducibility of under 73% with nearly half of the 21 scale positions unaccounted for.

### Discussion

The outcome of the scalogram analysis supports the idea that the progress of project teams from Level 1 to Level 2 is an orderable process. Level 2 practices are not acquired randomly. The strength of the extracted scale indicates the remarkable degree of commonality among Level 2 practices: they do indeed tap into a single underlying dimension as indicated by the fact that the full data set of 121 items, with a coefficient of reproducibility of 79%, almost

*Table 1.* The 46 scaleable items at Level 2 of the CMM.

| Item | Scale Score | Description |
|---|---|---|
| PT.CO.1 | 20 | A project software manager is designated to be responsible for the project's software activities and results. |
| QA.AB.1 | 20 | A group that is responsible for coordinating and implementing SQA for the project (i.e., the SQA group) exists. |
| QA.VE.3 | 20 | Experts independent of the SQA group periodically review the activities and software work products of the project's SQA group. |
| SM.CO.2 | 20 | A subcontract manager is designated to be responsible for establishing and managing the software subcontract. |
| PP.AC.1 | 19 | The software engineering group participates on the project proposal team. |
| PP.AC.3 | 19 | The software engineering group participates with other affected groups in the overall project planning throughout the project's life. |
| SM.VE.2 | 19 | The activities for managing the software subcontract are reviewed with the project manager on both a periodic and event-driven basis. |
| RM.VE.2 | 18 | The activities for managing the allocated requirements are reviewed with the project manager on both a periodic and event-driven basis. |
| PP.VE.2 | 18 | The activities for software project planning are reviewed with the project manager on both a periodic and event-driven basis. |
| PP.CO.1 | 17 | A project software manager is designated to be responsible for negotiating commitments and developing the project's software development plan. |
| PP.AB.2 | 16 | A documented and approved statement of work exists for the software project. |
| QA.VE.2 | 15 | The SQA activities are reviewed with the project manager on both a periodic and event-driven basis. |
| CM.VE.2 | 15 | The SCM activities are reviewed with the project manager on both a periodic and event-driven basis. |
| PT.VE.2 | 15 | The activities for software project tracking and oversight are reviewed with the project manager on both a periodic and event-driven basis. |
| CM.AB.2 | 15 | A group that is responsible for coordinating and implementing SCM for the project (i.e., the SCM group) exists. |
| PP.AC.6 | 14 | The project's software development plan is developed according to a documented procedure. |
| QA.AC.3 | 13 | The SQA group participates in the preparation and review of the project's software development plan, standards and procedures. |
| PT.AB.1 | 12 | A software development plan for the software project is documented and approved. |
| PT.AC.1 | 12 | A documented software development plan is used for tracking the software activities and communicating status. |
| SM.AC.4 | 11 | A documented subcontractor's software development plan is reviewed and approved by the prime contractor. |
| QA.AC.7 | 10 | Deviations identified in the software activities and software work products are documented and handled according to a documented procedure. |
| PP.AB.4 | 10 | The software managers, software engineers, and other individuals involved in the software project planning are trained in the software estimating and planning procedures applicable to their areas of responsibility. |
| CM.AC.3 | 10 | A configuration management library system is established as a repository for the software baselines. |
| CM.AC.7 | 10 | Products from the software baseline library are created and their release is controlled according to a documented procedure. |

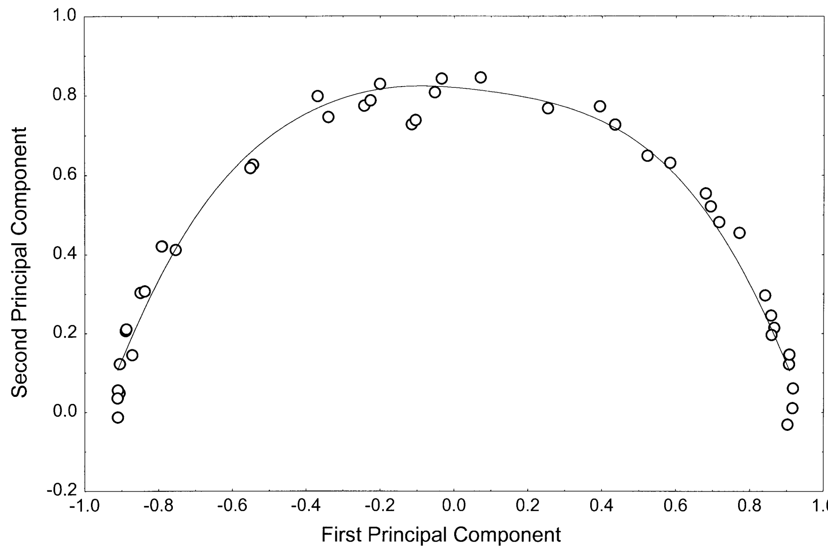


*Figure 5.* A scalogram chart of the 46-item Guttman scale extracted by this study. The coefficient of reproducibility for this scale is 93%. The uneven spacing of points along the pure scale line reflects the fact that some items still contain scaling error, although no more than one error per item. As in Figures 3 and 4, the scale reads clockwise with the most frequently satisfied items to the left and the least frequently satisfied items to the right.

meet the 80% criterion for an acceptable scale. Table 1 and Figure 4 both provide the order in which teams were able to implement successful practices. It provides an implementation strategy for movement from Level 1 to Level 2.

The 46 items that define the refined scale represent milestones in the attainment of Level 2. The scale was found by selecting those items with minimal scaling error, paring the scale down to its essentials. In scalogram analysis, the goal is to arrive at no more than one item at each position of the purified scale. In that sense, they act as markers indicating, step-by-step, the progress of a team toward Level 2. An implication of this is that a refined scale (a scale with only one item per possible scale position) can be used as a supplementary audit tool to take snapshot assessments of teams. The fact that the scaling item set extracted in this study includes more items than necessary to construct a complete scale means that parallel scales can be constructed for use in test-retest designs. It is important to keep in mind that the CMM itself requires that all 121 practices be implemented if a team is to be judged “Level 2.” What these milestone practices provide is a first-order approximation of the scale underlying the full item set.

*Table 1.* Continued.

| | | |
|---|---|---|
| PT.AC.2 | 10 | The project's software development plan is revised according to a documented procedure. |
| CM.AC.6 | 10 | Changes to baselines are controlled according to a documented procedure. |
| QA.AC.1 | 9 | A SQA plan is prepared for the software project according to a documented procedure. |
| PP.AC.7 | 9 | The plan for the software project is documented. |
| CM.AB.1 | 9 | A board having the authority for managing the project's software baselines (i.e., a software configuration control board—SCCB) exists or is established. |
| PP.AC.13 | 9 | The software risks associated with the cost resource, schedule, and technical aspects of the project are identified, assessed, and documented. |
| PT.AC.6 | 9 | The project's software effort and costs are tracked, and corrective actions are taken as necessary. |
| CM.AC.8 | 8 | The status of configuration items/units is recorded according to a documented procedure. |
| QA.AC.2 | 8 | The SQA group's activities are performed in accordance with the SQA plan. |
| SM.AC.10 | 7 | The prime contractor's software quality assurance group monitors the subcontractor's software quality assurance activities according to a documented procedure. |
| SM.AC.11 | 6 | The prime contractor's software configuration management group monitors the subcontractor's activities for software configuration management according to a documented procedure. |
| CM.AB.5 | 6 | Members of the software engineering group and other software-related groups are trained to perform their SCM activities. |
| RM.VE.3 | 6 | The software quality assurance group reviews and/or audits the activities and work products for managing the allocated requirements and reports the results. |
| SM.ME.1 | 5 | Measurements are made and used to determine the status of the activities for managing the software subcontract. |
| PP.AC.15 | 4 | Software planning data are recorded. |
| PT.AC.11 | 4 | Actual measurement data and replanning data for the software project are recorded. |
| QA.AC.5 | 3 | The SQA group audits designated software work products to verify compliance. |
| CM.VE.3 | 2 | The SCM group periodically audits software baselines to verify that they conform to the documentation that defines them. |
| PP.AC.9 | 2 | Estimates for the size of the software work products (or changes to the size of software work products) are derived according to a documented procedure. |
| CM.AC.10 | 2 | Software baseline audits are conducted according to a documented procedure. |
| CM.ME.1 | 0 | Measurements are made and used to determine the status of the SCM activities. |
| SM.VE.3 | 0 | The software quality assurance group reviews and/or audits the activities and work products for managing the software subcontract and reports the results. |

### *Interpreting the Underlying Dimension*

The search for a description of the underlying dimension of Level 2 leads us to an investigation of how the 46 scaleable practices are distributed across the KPAs or CFAs. Figure 6 shows the distribution of items across KPAs and CFAs. The 46 scale items are listed in descending order by scale score from most frequently satisfied to least frequently satisfied. Figures 7 and 8 provide scalogram charts for the individual KPAs and CFAs respectively.

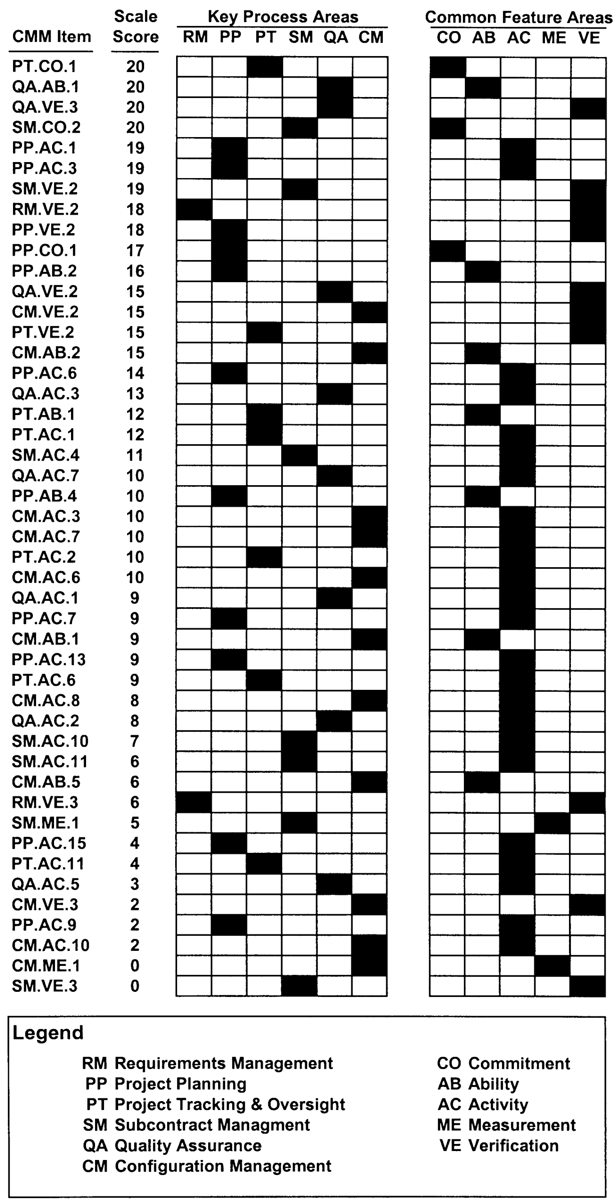


*Figure 6.* Distribution across KPAs and CFAs of the 46 scaleable items at Level 2 of the CMM.

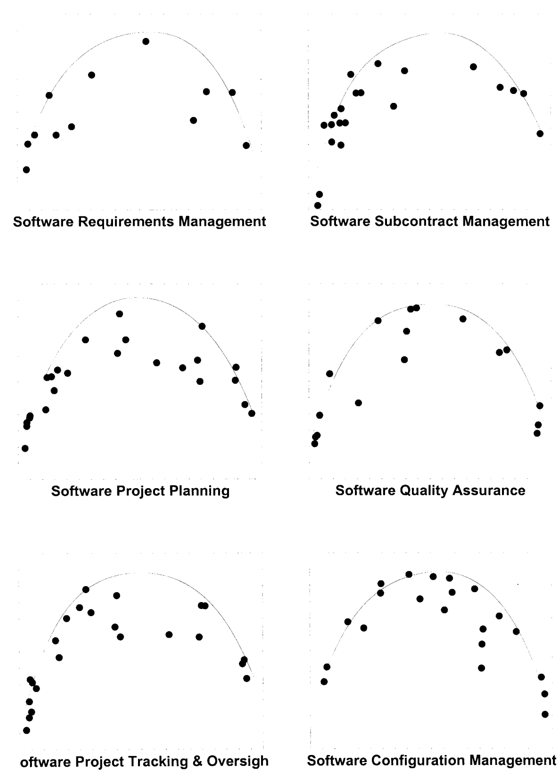


*Figure 7.* Item scaling properties of the KPAs at Level 2. The KPA items have been plotted relative to a pure Guttman scale.

It can be seen in Figure 6 that there is no systematic distribution pattern across KPAs. On the other hand, there is some structural linkage with the CFAs: Commitment and Ability items generally fall at the "most frequently satisfied" end of the scale and Measurement items generally fall at the "least frequently satisfied" end (see Figure 8.) The Verification items seem to have polarized at opposite ends of the scale while the Activity items are scattered across it. Although the CFA structure provides hints as to the meaning of the underlying dimension, it does not provide a clear or convincing interpretation because there is no consistent order among the CFAs. Ability falls at the left end, Measurement falls at the right, and the rest of the CFAs are scattered. Johnson and Brodman (1996) found similar asymmetries in the distribution of implemented activities. They found a growth pattern as well as a change in the types of metrics used as maturity increased. The question, then, becomes whether the underlying dimension enhances our understanding of how the key practices might best be implemented.

To answer this, the contents of the 46 scaleable items were sorted by one of the authors using an unconstrained Q-sort (Weller and Romney, 1988). In a Q-sort, the subject sorts

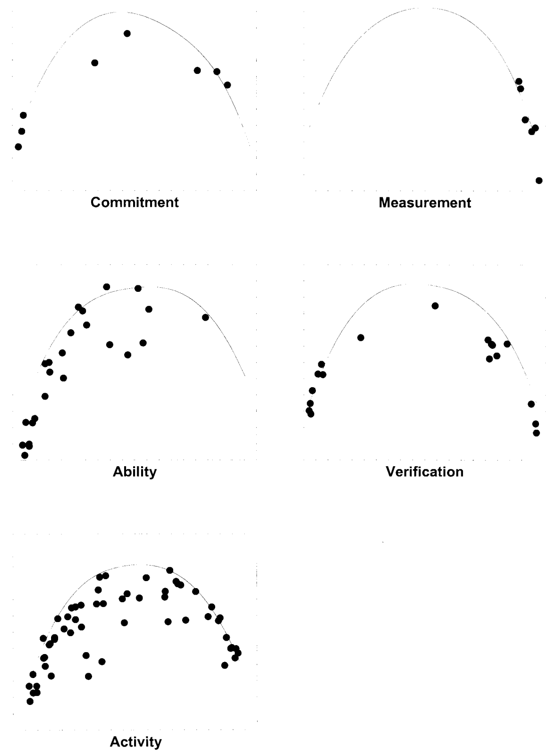


*Figure 8.* Item scaling properties of the CFAs at Level 2. The CFA items have been plotted relative to a pure Guttman scale.

the items into any number of categories based on perceptions of similarities in the verbal content. The same task was performed independently by a senior quality assurance manager familiar with the CMM (but not the results of the study). The number of categories and the placement of items within categories were identical for both sorters but one. One sorter subdivided one of the categories into two. The classifications were compared and the sorters agreed that there was no distortion in collapsing the subdivided categories into one. This resulted in six categories covering 42 of the 46 items. They six categories are "designation of roles and responsibilities," "project management oversight," "planning document development," "use of documented procedures," "measurement of activities and outcomes," and "independent review of product and process quality." The remaining 4 items were independently designated as unclassifiable by both sorters. The categories and the number of items falling under each are listed in Table 2. Figure 9 provides an alternative view of the same information, a scalogram chart of the 46 items labeled by category.

This offers a somewhat cleaner classification scheme for the scale items than either the KPAs or the CFAs because each of the six falls in a different location in the Guttman scale.

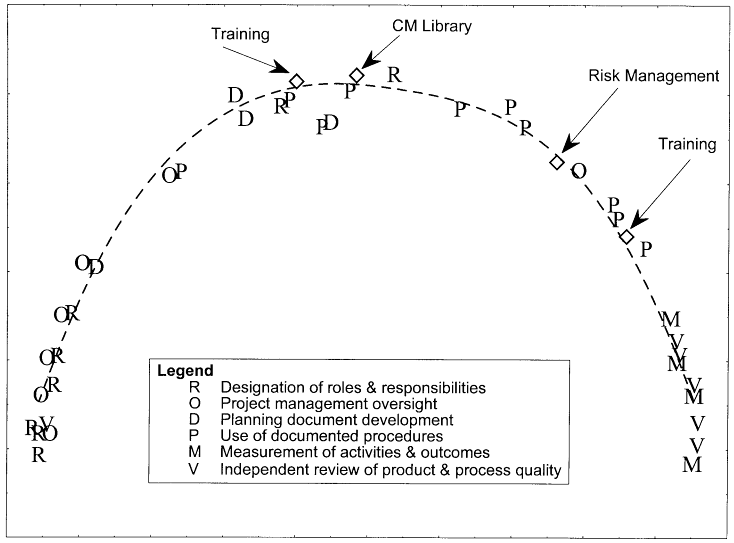


*Figure 9.* Distribution of the 46 scale items identified in this study across the classifications proposed by the authors. The 4 unclassified items are identified by diamonds.

Under this classification scheme, the extracted scale implies that projects progress toward Level 2 in roughly six steps. The four scale items that did not fit into the classification categories are not trivial: two relate to training (PP.AB.4 and CM.AB.5), one to the use of automated tools (CM.AC.3), and one to formal risk assessment (PP.AC.13).

What is clear is that this partitioning of the scale items resembles the classic PDCA cycle of Plan-Do-Check-Act. "Designation of roles and responsibilities" and "Planning document development" correspond to the "Plan" step of the cycle. "Use of documented procedures" corresponds to the "Do" step. "Project management oversight", "Measurement of activities and outcomes", and "Independent review of product and process quality" correspond to the "Check" step. The "Act" step of the PDCA cycle is embedded in the "Check" step owing to the way in which many CMM items are written. For example, PT.AC.6 reads "The project's software effort and costs are tracked, and corrective actions are taken as necessary." In a single item, the model is addressing both measurement (Check) and corrective action (Act).

The relationship between the behavior of real-world teams trying to implement the CMM and the PDCA cycle is not too surprising given that the PDCA cycle is such an intuitive strategy for managers. There is no conflict in this relationship either. Instead, it implies that a practical strategy for implementation exists. Further analysis with more diverse project teams may clarify the nature and placement of the categories as well as the role of the unclassified items.

If the results of the scalogram analysis and the PDCA cycle point us in the direction of a practical implementation strategy for Level 2, then why don't more teams make it to

*Table 2.* An alternative classification scheme for the 46 scaleable items at Level 2 of the CMM.

| Category | No. of Items | Items | | |
|---|---|---|---|---|
| Designation of roles and responsibilities | 9 | PT.CO.1 | QA.AB.1 | SM.CO.2 |
| | | PP.AC.1 | PP.AC.3 | PP.CO.1 |
| | | CM.AB.2 | QA.AC.3 | CM.AB.1 |
| Project management oversight | 8 | SM.VE.2 | RM.VE.2 | PP.VE.2 |
| | | QA.VE.2 | CM.VE.2 | PT.VE.2 |
| | | PT.AC.1 | PT.AC.6 | |
| Planning document development | 4 | PP.AB.2 | PT.AB.1 | SM.AC.4 |
| | | PP.AC.7 | | |
| Use of documented procedures | 10 | PP.AC.6 | QA.AC.7 | CM.AC.7 |
| | | PT.AC.2 | CM.AC.6 | QA.AC.1 |
| | | CM.AC.8 | QA.AC.2 | SM.AC.10 |
| | | SM.AC.11 | | |
| Measurement of activities and outcomes | 5 | SM.ME.1 | PP.AC.15 | PT.AC.11 |
| | | PP.AC.9 | CM.ME.1 | |
| Independent review of product and process quality | 6 | QA.VE.3 | RM.VE.3 | QA.AC.5 |
| | | CM.VE.3 | CM.AC.10 | SM.VE.3 |
| Unclassified | 4 | PP.AB.4 | CM.AC.3 | PP.AC.13 |
| | | CM.AB.5 | | |

Level 2? In fact, there is a core conflict at play. Project teams are often charged with implementation of the key practices and delivery of the product at the same time. Without a clear understanding of a practical implementation strategy, however, they flounder trying to decide how to start or they are paralyzed by the enormity of it. Even if key practices can be implemented, they are not always done in the optimal sequence which means they are less likely to "stick." Practices that do not add unequivocal value will be discarded and the team will focus exclusively on the product.

The evolutionary viewpoint presented earlier can help us understand when a practice will "stick" and when it will not. As stated earlier, to survive, each key practice requires a project environment that is minimally capable of sustaining it. Practices which are introduced into "prepared" environments survive because the prerequisite practices have been implemented. Unprepared practices die out—they may be implemented but they will not become institutionalized.

### *Limitations of the Study*

This study is an exploratory analysis of the scaling properties of key practices within one level of the CMM based on the experiences of ten small software development teams. The KPA-based organization of Level 2 reflects the large project origins of the Capability Maturity Model. The KPAs represent commonly-used divisions of work across large organizations: one group develops requirements, another group handles configuration management, and so on. In small project teams, however, individuals must perform many roles and many roles must be shared. The structure of our extracted scale, cutting across KPAs, may in fact reflect the way small project team members share responsibilities.[1] Clearly, there is a need for replication on a larger and more diverse sample of project teams. The scaling properties of Levels 3, 4, and 5 need to be examined as well.

For this study, scalogram analysis can be an effective tool in the development of a theory but is not an inferential test of one. A rigorous experimental design is difficult to do on subjects of this type because it would most likely have to be done on "toy" teams, making the results difficult to generalize. Naturalistic observation is used instead and, for that reason, the quality of a behavioral theory developed from scalogram analysis can only be inferred through the accumulated evidence of confirming studies (Borg and Shye, 1995).

## Notes

1. Our appreciation to Janet Deeney, SBC Services, Inc., for raising this issue.

## References

Borg, I., and Shye, S. 1995. *Facet Theory: Form and Content*. Thousand Oaks, CA: Sage Publications.

Brodman, J. G., and Johnson, D. I. 1996. Return on investment from software process improvement as measured by U.S. industry. *Crosstalk* 9(4).

Gorden, R. L. 1977. *Unidimensional Scaling of Social Variables: Concepts and Procedures*. New York: The Free Press.

Guttman, L. L. 1944. A basis for scaling qualitative data. *American Sociological Review* 9: 139–150.

Johnson, D. I., and Brodman, J. G. 1996. Realities and rewards of software process improvement. *IEEE Software* 29(11): 99–101.

Johnson, D. I., and Brodman, J. G. 1997. Tailoring the CMM for small businesses, small organizations, and small projects. *Software Process Newsletter* 8: 1–6.

Lawlis, P. K., Flowe, R. M., and Thordahl, J. B. 1995. A correlational study of the CMM and software development performance. *Crosstalk* 8(9).

McIver, J. P., and Carmines, E. G. 1981. *Unidimensional Scaling*. Thousand Oaks, CA: Sage Publications.

Paulk, M. C., Curtis, B., Chrissis, M. B., and Weber, C. V. 1993. *Capability Maturity Model for Software*, Version 1.1. Pittsburgh, PA: Software Engineering Institute.

Piaget, J., and Inhelder, B. 1969. *The Psychology of the Child*. New York: Basic Books.

Shye, S. 1978. *Theory Construction and Data Analysis in the Behavioral Sciences*. San Francisco: Jossey-Bass Publishers.

Weller, S. C., and Romney, A. K. 1990. *Metric Scaling: Correspondence Analysis*. Thousand Oaks, CA: Sage Publications.

Weller, S. C., and Romney, A. K. 1988. *Systematic Data Collection*. Thousand Oaks, CA: Sage Publications.

**John G. Bilotta** is currently director of change management services for Charles Schwab & Co., Inc. He has also worked at Pacific Telesis, Blue Shield of California, and Bank of America as a systems analyst, software developer, technical lead, quality assurance manager, and system engineer. He received his Masters and Bachelors degrees from the University of California at Berkeley with an emphasis in psychological measurement and individual differences. He is a member of the Association for Computing Machinery, the IEEE Computer Society, and the Quality Assurance Institute.

**John F. McGrew** is currently principal research engineer for SBC Services, Inc., with responsibility for applied research in strategic system architectures, management decision making, and user computer interactions. His previous experience is in human factors engineering, artificial intelligence, training effectiveness, and cognitive and experimental psychology. He has also worked at the Jet Propulsion Laboratory and the Army Research Institute for the Social and Behavioral Sciences. He earned his Ph.D. at the University of Georgia, and his Masters and Bachelors degrees at California State University at Fresno. He belongs to the ACM Computer Human Interaction Society, the IEEE Computer Society, the American Association for Artificial Intelligence, and the Human Factors Society.