The Internal Consistencies of the 1987 SEI Maturity Questionnaire and the SPICE Capability Dimension[*]

Pierfrancesco Fusaro
Fraunhofer Institute for Experimental
Software Engineering
Sauerwiesen 6
D-67661 Kaiserslautern
Germany
fusaro@iese.fhg.de

Khaled El Emam
Fraunhofer Institute for Experimental
Software Engineering
Sauerwiesen 6
D-67661 Kaiserslautern
Germany
elemam@iese.fhg.de

Bob Smith
European Software Institute
Parque Tecnologico de Zamudio #204
E-48170 Bilbao
Spain
Bob.Smith@esi.es

Abstract

This paper presents the results of an empirical evaluation of the reliability of two commonly used assessment instruments: the 1987 SEI maturity questionnaire and the SPICE v1 capability dimension. The type of reliability that was evaluated is internal consistency. A study of the internal consistency of the 1987 questionnaire was only briefly mentioned in a 1991 article, and the internal consistency of the SPICE v1 capability dimension has not been evaluated thus far. We used two different data sets to evaluate the internal consistency of each instrument. Our results indicate that both assessment instruments are very reliable and also have similar reliability levels. The results are encouraging for users of assessment instruments, and provide a baseline with which to compare subsequent versions of these instruments.

1. Introduction

One of the recently developed class of decision making tools for software engineering management is the software process assessment. The scores from such assessments are being applied in focusing and tracking self-improvement efforts, and as part of the contract award decision making process. When ratings are assigned to measure the maturity of organizations or the capability of processes, during or as a consequence of an assessment, then an assessment can be considered as a measurement procedure. Furthermore, because assessment ratings depend upon the judgment of assessors, they are a subjective measurement procedure.

Given the importance of the decisions made by organization based on assessment scores, the issue of the reliability of assessments is very critical. Reliability is defined in general as the extent to which the same measurement procedure will yield the same results on repeated trials [3]. Lack of reliability is caused by random measurement error. For subjective measurements, there are many potential sources of random measurement error. In the context of process assessment, some potential sources of error have been described in [9].

A lack of reliability in assessments may lead to erroneous decisions when one compares assessment scores. For example, comparisons occur when deciding which is the weakest process to invest resources in, when comparing the ratings of potential suppliers, or when tracking ratings over time to monitor process improvement progress.

The objective of this paper is to report on a study of the reliability of two assessment instruments: the 1987 SEI maturity questionnaire, and the SPICE capability dimension. The method of evaluating reliability uses an internal consistency coefficient, the same as the one reported by Humphrey and Curtis [18] and used in [10].

One of the earliest assessment instruments was the maturity questionnaire developed at the SEI [17]. Initial assessments used this questionnaire in the process of scoring the software engineering practices of organizations [16]. More recently the CBA-IPI [8] and the Interim Profile [32] assessment

[*] This paper appears as International Software Engineering Research Network technical report ISERN-97-01.

methods were developed. In the context of source selection, the assessment method is known as the Software Capability Evaluation [27].

While the SEI has released a new maturity questionnaire in 1994 [33], it is of scientific and practical utility to evaluate the reliability of the 1987 maturity questionnaire for the following three reasons:

- it is a scientific necessity to replicate and confirm the results obtained by other researchers [22], and ours can be considered a confirmatory study of the one mentioned in [18] where the internal consistency of the 1987 questionnaire is briefly mentioned.
- evaluation of the 1987 maturity questionnaire serves as a baseline for comparing subsequent improvements to the questionnaire[1]; if there is no baseline, then future studies of the reliability of the 1994 maturity questionnaire would not be able to indicate whether it was an improvement or deterioration;
- the 1987 maturity questionnaire is still being used; for example, it was used for a survey conducted in Singapore on the maturity of organizations [31] and up until the November 1996 issue of the maturity profile of the software industry [29], data collected since 1987 using the 1987 maturity questionnaire was still being presented and included in the profile; therefore data using the 1987 maturity questionnaire is still being applied in characterizing the maturity status of the software industry in the US and internationally.

The SPICE project aims to deliver an ISO standard on software process assessment [26]. This project builds upon the experiences gained with contemporary assessment models. As part of this project there are a set of empirical trials [23]. During the trials, data has been collected from early users of the assessment framework.

Briefly, our results indicate that the reliability (calculated using an internal consistency method described later in the paper) of both the 1987 maturity questionnaire and the SPICE v1 capability dimension is very high. Furthermore, both have similar values. These results are encouraging for users of software process assessments. They also establish a baseline which can be used in the evaluation of future versions of the questionnaire and the capability dimension, and for comparison with other contemporary assessment instruments.

In this paper we first review basic reliability concepts and previous work on the reliability of assessments. We then describe the data sets used in this study, and we explain the data analysis method that we followed. In section 4 we present the results of the evaluation of the internal consistency of the 1987 version of the SEI maturity questionnaire and the SPICE v1 capability dimension, and discuss the results achieved in comparison with those obtained in [18]. Section 5 concludes the paper with a summary and directions for future research.

2. Background

2.1 The Assessment Instruments

The reliabilities of two assessment instruments[2] were evaluated. These two instruments measure different things: one organizational maturity and one process capability. The difference between these two general approaches used in process assessments is explained in [25]. Below we give only an overview of the ratings employed by the two approaches.

The first, the 1987 SEI maturity questionnaire consists of 85 questions covering software engineering practices and 16 technology questions. The questionnaire measures the maturity of organizations. As noted in [2], the technology questions are not graded to determine the maturity of an organization. Organizational maturity is measured on a five point scale indicating extent of maturity. The grading

[1] In [18] only a note of the results is given. The study's details and assumptions are not presented.

[2] In this paper we use the term instrument to mean a questionnaire or any other form of data collection mechanism that allows rating of software engineering practices against an assessment model. For example, this can be achieved using automated tools.

algorithm to get from the responses to the 85 questions to a score on the 5-point maturity scale is described in [2].

The maturity questionnaire is used in two contexts. First, it is used as a stand-alone tool for evaluating the maturity of organizations. Second, it is used as an orientation tool during an assessment.

The purpose of using the questionnaire as a stand-alone tool has commonly been to produce a profile of the maturity of organizations. Data is collected from a sample that is assumed to be representative of a larger population. An example is the study conducted by Humphrey et al. [19] on the maturity of Japanese software companies. The bulk of this data was collected from large meetings where professionals rate their own organizations (similar to assessment tutorials). Another more recent example is the survey conducted to characterize the maturity of software organizations in Singapore [31].

It has been recommended that the maturity questionnaire should not be used as a stand-alone assessment tool, but rather as an orientation tool during an assessment and to establish an initial maturity level [16]. In such a context, the questionnaire is filled during the preparation phase of the assessment and on the first day of the assessment phase [16]. An example of this kind of application is given in [28] where the questionnaire was part of a request for proposal in a source selection situation.

The reliability of the maturity questionnaire is an important issue in both contexts described above. A low reliability questionnaire may produce an erroneous characterization of organizational maturity and incorrect initial maturity scores, which may lead to inefficient assessments due to misorientation. It is expected, however, that an empirically estimated reliability of the questionnaire from data collected during assessment tutorials or assessments would be higher than from a survey because in the former contexts the respondents have the opportunity to clarify any ambiguities.

The unit of analysis in the SPICE framework [20] is a process instance. A process instance is defined as “a singular instantiation of a process that is uniquely identifiable and about which information can be gathered in a repeatable manner” [21]. The capability of process instances is rated during assessments. The SPICE architecture has two dimensions: a capability and a process dimension. The capability of processes is evaluated using 26 generic practices that span 6 capability levels. Each successive level indicates a higher process capability. More details of the general SPICE architecture are given in Appendix A.

Ratings in a SPICE-based assessment are usually made on a form. The exact nature of this form is not stipulated in the SPICE documents. However, the definitions of the generic practices against which one does the ratings, the processes that are rated, and the rating scheme are explicitly defined in the documents. Therefore, an instrument in this case consists of the generic practices and the rating scheme that are defined in the SPICE documents.

2.2 The Reliability of Assessments

The two approaches for evaluating reliability that have been studied in a process assessment context are the internal consistency of assessment instruments (e.g., see [11][18]) and interrater agreement (e.g., see [12][15]). Internal consistency measures the extent to which the components of an instrument have been constructed to the same or to consistent content specifications of what the instrument is supposed to measure. It is also affected by ambiguities in wording and inconsistencies in interpretation by respondents. This approach accounts for consistency within the set of rating scales as a source of measurement error. Interrater agreement measures the extent to which independent assessors produce the same ratings when presented with the same evidence. This approach accounts for the different assessors as a source of measurement error. In the current study we are only concerned with internal consistency.

2.3 Basic Concepts

A basic concept for comprehending the reliability of measurement is that of a *construct*. A construct refers to a meaningful conceptual object. A construct is neither directly measurable nor observable. However, the quantity or value of a construct is presumed to cause a set of observations to take on a certain value. An observation can be considered as a question in a maturity questionnaire (this is also referred to as an *item*). Thus, the construct can be indirectly measured by considering the values of those items.

For example, organizational maturity is a construct. Thus, the value of an item measuring *"the extent to which projects follow a written organizational policy for managing system requirements allocated to software"* is presumed to be caused by the true value of organizational maturity. Also, the value of an item measuring *"the extent to which projects follow a written organizational policy for planning software projects"* is presumed to be caused by the true value of organizational maturity. Such a relationship is depicted in the path diagram in Figure 1. Since organizational maturity is not directly measurable, the above two items are intended to estimate the actual magnitude or true score of organizational maturity.

Since reliability is concerned with random measurement error, error must be considered in any theory of reliability. The classic theory (see [1]) states that an observed score consists of two components, a true score and an error score: X = T + E. Thus, X is the score obtained in a maturity assessment, T is the mean of the theoretical distribution of X scores that would be found in repeated assessments of the same organization using the same maturity assessment procedure, and E is the error component.

The true score is considered to be a perfect measure of maturity. In practice, however, the true score can never be really known since it is generally not possible to obtain a large number of repeated assessments of the same organization. True scores are therefore only hypothetical quantities, but useful nevertheless.

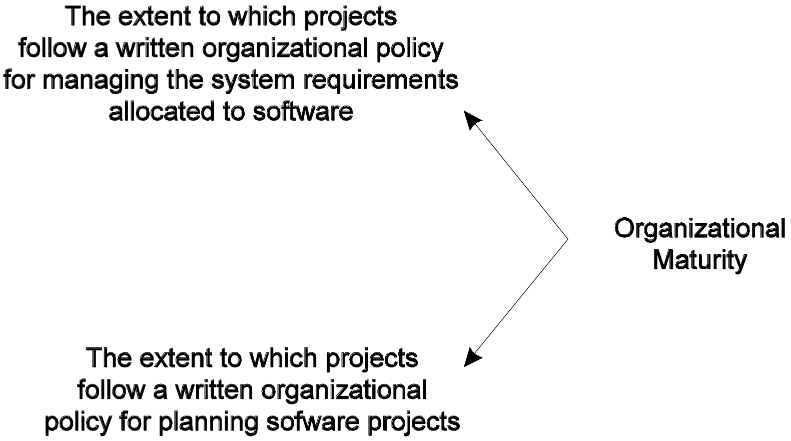
The extent to which projects follow a written organizational policy for managing the system requirements allocated to software
Organizational Maturity
The extent to which projects follow a written organizational policy for planning sofware projects

Figure 1: Path diagram depicting the organizational maturity construct and example items for its measurement.

2.4 Estimating Reliability

The most commonly used estimate of internal consistency is the Cronbach alpha coefficient [4]. This is the same coefficient that has been used by Humphrey and Curtis [18] in their note about the internal consistency of the 1987 maturity questionnaire, and in a recent study of the internal consistency of an organizational maturity instrument [10]. Below we describe the logic behind this coefficient. We use organizational maturity as the construct being measured for ease of presentation, but the explanations are equally valid for the process capability construct.

The type of scale used in the most common assessment instruments is a summative one. This means that the individual scores for each item are summed up to produce an overall score. One property of the covariance matrix for a summative scale that is important for the following formulation is that the sum of all the elements in the matrix give exactly the variance of the scale as a whole.

One can think of the variability in a set of item scores as being due to one of two things: (a) actual variation across the organizations in maturity (i.e., true variation in the construct being measured) and this can be considered as the signal component of the variance, and (b) error which can be considered as the noise component of the variance. Computing the Cronbach alpha coefficient involves partitioning the total variance into signal and noise. The proportion of total variation that is signal equals alpha.

The signal component of variance is considered to be attributable to a common source, presumably the true score of the construct underlying the items. When maturity varies across the different organizations, scores on all the items will vary with it because it is a cause of these scores. The error terms are the source of unique variation that each item possesses. Whereas all items share variability due to maturity, no two items share any variation from the same error source.

Unique variation is the sum of the elements in the diagonal of the covariance matrix: $\Sigma\sigma_i^2$. Common variation is the difference between total variation and unique variation: σ_y^2 - $\Sigma\sigma_i^2$, where the first term is the variation of the whole scale. Therefore, the proportion of common variance can be expressed as: $(\sigma_y^2 - \Sigma\sigma_i^2)/\sigma_y^2$. To express this in relative terms, the number of elements in the matrix must be considered. The total number of elements is N^2, and the total number of elements that are communal are N^2 - N. Thus the corrected equation for coefficient alpha becomes:

$$\alpha = \frac{N}{(N-1)}\left[1 - \sum \sigma_i^2 / \sigma_y^2\right]$$

The Cronbach alpha coefficient varies between 0 and 1. If there is no true score but only error in the items, then the variance of the sum will be the same as the sum of variances of the individual items. Therefore, coefficient alpha will be equal to zero (i.e., the proportion of true scores in the scale is zero percent). If all items are perfectly reliable and measure the same thing, then coefficient alpha is equal to one (i.e., the proportion of true score in the scale is 100 percent).

Cronbach's alpha is a generalization of a coefficient introduced by Kuder and Richardson to estimate the reliability of scales composed of dichotomously scored items. Dichotomous items are scored one or zero depending on whether the respondent does or does not endorse the particular characteristic under investigation. To determine the reliability of scales composed of dichotomously scored items, the Kuder-Richardson formula (symbolized KR20) is [1]:

$$\mathrm{KR20} = \frac{\mathrm{N}}{(\mathrm{N}-1)}\left[1 - \Sigma \mathrm{p_i q_i} / \sigma_\mathrm{x}^2\right]$$

where N is the number of dichotomous items; p_i is the proportion responding positively to the i^{th} item; q_i is equal to $1 - p_i$; and σ_y^2 is equal to the variance of the total composite. Since KR20 is simply a special case of alpha, and it has the same interpretation as alpha.

2.5 Predicting Reliability for a Different Length Instrument

When one is not able to collect data on all of the items of an assessment instrument[3], it is still possible to evaluate the reliability for all of the questions in the instrument. A well known formulation for doing so is the Spearman-Brown prophecy formula:

$$SB = \frac{k \times KR20}{1 + ((k-1) \times KR20)}, \text{ or } SB = \frac{k \times \alpha}{1 + ((k-1) \times \alpha)}$$

where k is the number of times the instrument must be larger or smaller than the current one. For example, for an instrument twice the current size, k=2. Usually, reliability increases when new questions are added. However, as also noted in [18], adding new questions may actually reduce reliability if the new questions correlate poorly with the other already existing questions.

[3] Or alternatively when one wishes to estimate the reliability after increasing/reducing the size of an existing instrument.

The Spearman-Brown formula can be used to facilitate comparisons between instruments. As noted in [1] when comparing two instruments the longer instrument will in general have a higher reliability because it is longer. Therefore, it would be appropriate to estimate the reliability for a different length instrument to make the instruments being compared of equal length.

2.6 Interpreting Reliability Coefficients

There are two approaches that can be used to determine whether an internal consistency value is good or bad. The first is by reference to some commonly accepted threshold. Since internal consistency thresholds have not been established in software engineering, we can resort to using general guidelines from other disciplines, namely psychometrics. The second is by reference to values obtained in other studies of assessment instruments conducted in the software engineering domain.

What a satisfactory level of reliability is depends on how a measure is being used. In the early stages of the research on assessment instruments, reliabilities of 0.7 or higher are considered sufficient. For basic research, a value of 0.8 is acceptable. However, in applied settings where important decisions are made with respect to assessment scores, a reliability of 0.9 is the minimum that would be acceptable [24].

To our knowledge, only two studies have been published in the past that evaluate the internal consistency of instruments used in assessing the maturity of organizations or the capability of software processes. These are presented below.

In their 1991 critique of the SEI's Software Capability Evaluations, Bollinger and McGowan [2] question the 'statistical reliability' of the algorithm used to calculate the maturity of organization. In their reply, Humphrey and Curtis note that some reliability studies of this questionnaire were conducted using "data gathered from SEI assessments" and quote a figure of 0.9 for the level 2 and level 3 questions [18]. Levels 2 and 3 contain 65 questions. If we use the Spearman-Brown formula above, we can estimate that the full 85 question maturity questionnaire has a reliability of 0.92. This is larger then the commonly accepted minimal threshold value of 0.9 for applied settings [24].

The second study in [10] evaluated the internal consistency of a questionnaire that covered four generic dimensions of organizational maturity: standardization, project management, tools, and organization. The overall internal consistency value for the composite of the 4 dimensions was approximately 0.95, a value sufficiently high for applied settings [24].

Therefore, in general, an absolute minimum value of internal consistency that would be acceptable is 0.8. However, an achievable value that is more desirable for applied settings would be 0.9. Since in our study, the maturity questionnaire and the SPICE capability dimension are both being used in making important decisions, the minimal tolerable value of internal consistency for these instruments should be set at 0.9.

3. Research method

3.1 Maturity Questionnaire Data Source

We used two data sets each representing the responses of two samples of practicing maintainers. This is the same data set used in [7]. The first sample consisted of forty surviving respondents from an original base of sixty-two who were members of a third-round Delphi panel in a large study conducted by Dekleva [6]. The second sample of subjects consisted of thirty nine respondents who completed a mail survey designed specifically for measuring maturity.

Each subject was an experienced software maintainer with an average of 15 years in information systems of which 11 years were spent in software maintenance or management of maintenance and therefore was qualified as an appropriate subject. The first sample were solicited from persons who attended the 1990 and 1991 Software Management Association professional conferences and the second sample were attendees of the 1992 meeting of the same conference. Respondents to the questionnaire were rating the practices in their Information Systems departments.

The first data set consisted of the responses of 40 subjects to 25 maturity-related questions (i.e. 12 items associated with “repeatable, level 2”, and 13 items associated with “defined, level 3”). The second data set listed the responses of 39 subjects to 56 maturity-related questions (i.e., 21 items associated with “repeatable, level 2”, 23 items associated with “defined, level 3”, and 12 items associated with “managed, level 4”). More details of these questions are given in Appendix B.

3.2 SPICE Capability Dimension Data Source

We used two separate data sets for this study. Each is described below.

The first data set was obtained from three assessments conducted within European during 1996. In total, 50 process instances in 14 projects were assessed. Five of the projects were maintenance and the remainder new development. We analyzed the generic practices for levels 2 and 3.

The method used for the three assessments was as follows. First, there was a half day pre-assessment meeting between the assessors and the organizational unit personnel for introductions and scoping of the assessment. The first half day of the actual assessment consisted of an introduction to SPICE and to the assessment for all of the assessments participants. This is followed by two and a half days of information gathering and process ratings. Information was gathered for each process to be assessed through interviews and document reviews. Right after, the ratings for that process were made by the assessors. This is followed by a half day preparation of the final ratings and a meeting with the assessment sponsor. The assessment is then closed by a 2 hour presentation of the results of the assessment.

The second data set was obtained during the first phase of the SPICE trials [14]. During the first phase of the trials data was collected from 35 assessments, 20 of which were conducted in Europe, 1 in Canada, and 14 in the Pacific Rim. In total we have data from 314 assessed process instances. We obtained data about the assessment instrument used and the data collection method during the assessment from 17 assessors who conducted assessments in the phase 1 trials. We analyzed the generic prcatices for levels 1,2, and 3.

During the trials assessments, the most commonly used type of assessment instrument was a paper based checklist followed by a computerized spreadsheet (Figure 2). Apart from the spreadsheet, it was rare that any other form of computerized instrument was used. The instruments that were used were developed mostly by the assessors themselves (Figure 3). Very few of the assessors (only 35%) used the exemplar automated instrument provided by the SPICE project. Most of the information that was collected during the assessments was through interviews (Figure 4). Very few assessors used assessee self-reports (0%) or collected data prior to the on-site visit (12%).

Type of Assessment Instrument	Percentage of Assessors
Computer Based Flat File	(0/17) = 0%
Computerized Checklist	(0/17) = 0%
Computerized Expert System	(0/17) = 0%
Computerized Questionnaire	(1/17) = 6%
Computerized Relational Database	(1/17) = 6%
Computerized Scoring	(5/17) = 29%
Paper Based Questionnaire	(7/17) = 41%
Computerized Spreadsheet	(10/17) = 59%
Paper Based Checklist	(11/17) = 65%

Figure 2: Type of assessment instruments used by the assessors.

Developer(s) of the Assessment Instruments Used	Percentage of Assessors
Organizational Unit representative(s)	(0/17) = 0%
Third party tool builders / vendors	(1/17) = 1%
Supplied as an exemplar from the SPICE project	(6/17) = 35%
Experienced assessors(s)	(15/17) = 88%

Figure 3: The developer(s) of the assessment instruments that were used.

Method for Collecting Information	Percentage of Assessors
Assessee self reports	(0/17) = 0%
Data collection prior to on-site visits	(2/17) = 12%
Group Feedback sessions	(6/17) = 35%
Document or interim work product reviews	(7/17) = 41%
Interviews	(17/17) = 100%

Figure 4: The method(s) that the assessors used to collect information during the assessment.

3.3 Data Analysis

In calculating the internal consistency of the two instruments, we used the Cronbach alpha and the KR20 formulas for estimation, and the SB formula for projecting the theoretical reliability if the instruments are lengthened or shortened. However, one issue that requires further consideration during data analysis is dimensionality of the organizational maturity and process capability constructs.

The coefficients of internal consistency that we use assume that the construct being measured is unidimensional [1]. There are a number of assumptions that one can make about the dimensionality of the organizational maturity and process capability constructs to satisfy this requirement. It is not clear which assumption was made in [18] however.

The first assumption that can be made is that organizational maturity itself is a unidimensional construct, and therefore all of the questions in a maturity questionnaire are measuring this single construct (this is the assumption made in Figure 1). The alternative assumption is that organizational maturity is a multidimensional construct. This assumption has received some support from the point of view that treating maturity as a single dimension is an oversimplification of the factors that affect software development [30] and therefore represents a rather coarse measure of the implementation of software engineering practices [13].

If we assume that organizational maturity is a multidimensional construct, then there are alternative ways that one can conceptually differentiate amongst the subdimensions. One approach would be to treat each maturity level as a subdimension. Therefore, for example, there would be a subdimension “level 2 practices” that covers practices differentiating between the *Initial* level and the *Repeatable* level, and a subdimension “level 3 practices” that differentiates between the *Repeatable* level and the *Defined* level, and so on. However, this dimensional structure has not been empirically evaluated. Another approach is to empirically derive the subdimensions of maturity using an exploratory factor analysis, as done in [5].

In our analysis, we calculate the internal consistency of the maturity questionnaire assuming that maturity is a unidimensional construct. The primary reason for this is that, while the dimensionality of the maturity and capability constructs has been acknowledged, no systematic investigations of this dimensionality have been conducted thus far.

4. Results

4.1 The 1987 SEI Maturity Questionnaire

Assuming that maturity is a unidimensional construct, we calculated coefficient KR20 for each of the two data sets. The results are presented in Figure 5. These results show quite large reliability values. We would expect the value from data set 1 to be lower than the value from data set two because of the smaller number of items in data set 1.

Our data set does not include all of the questions that are in the original 1987 questionnaire. We can plot the number questions against the reliability, given that reliability will vary with questionnaire length. This is shown in Figure 6. As can be seen, the minimal value of a reliability of 0.9 is passed at 45 items for both data sets. This is a remarkably consistent result obtained from our two data sets. For the total questionnaire size of 85 items the value would be approximately 0.94 for both data sets. Compared to the estimated value of an 85 item questionnaire from the results in [18] (see Section 2.6) of 0.92, our obtained value is larger but consistent.

Given that our data was collected using a survey, it is expected that the internal consistency values calculated here are lower than what would be obtained from data collected during an assessment tutorial or an actual assessment. However, the values reported in [18] that are based on SEI assessments do not confirm that expectation, as shown in the above paragraph.

Data Set 1		**Data Set 2**	
# Items	**KR-20**	**# Items**	**KR-20**
25	0.84	56	0.92

Figure 5: Results from the internal consistency analysis for the 1987 SEI maturity questionnaire.

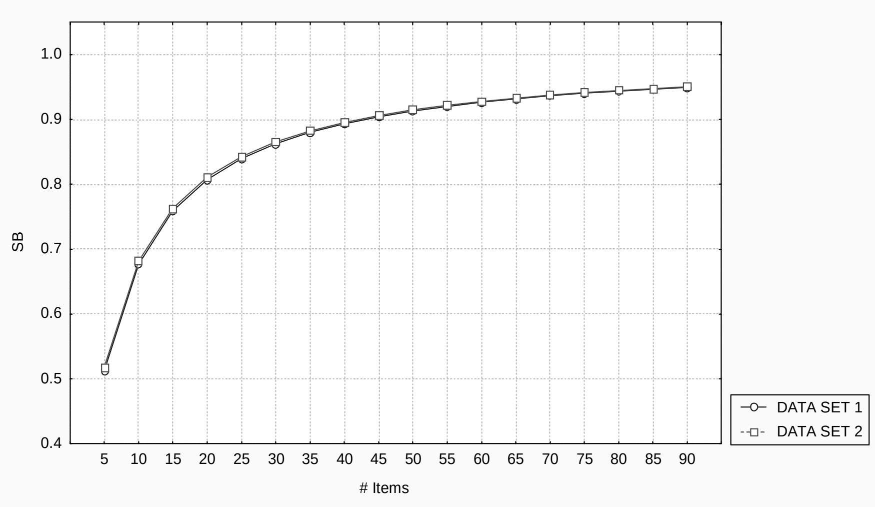

Figure 6: Reliability values of different questionnaire lengths for the two data sets.

4.2 The SPICE v1 Capability Dimension

We calculated the internal consistency of the SPICE capability dimension. The results are summarized in Figure 7. Total number of generic practices on the capability dimension is 26. The graph in Figure 8 shows the estimated reliability for different instrument lengths. It is clear that the reliability would increase to approximately 0.94 for the first data set and 0.97 for the second data set with a 26 item instrument. In general, these two values are consistent with each other.

Data Set 1		Data Set 2	
# Items	**Alpha**	**# Items**	**Alpha**
17	0.91	18	0.96

Figure 7: Results from the internal consistency analysis for the SPICE v1 capability dimension.

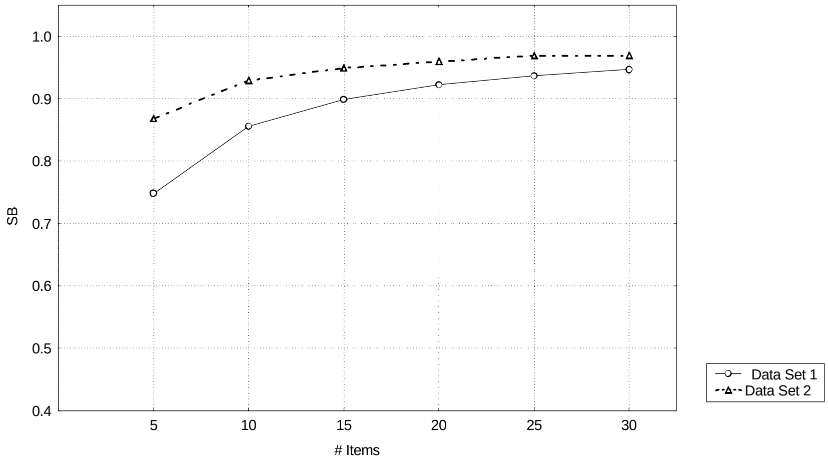

Figure 8: Reliability against the number of generic practices.

	1987 maturity questionnaire Data Set 1	1987 maturity questionnaire Data Set 2	Estimates based on [18]	SPICE v1 Capability Dimension Data Set 1	SPICE v1 Capability Dimension Data Set 2
85 item instrument	0.94	0.94	0.92	0.98	0.99
26 item instrument	0.84	0.84	0.78	0.94	0.97

Figure 9: Comparisons of results.

A comparison of the SPICE results with the maturity questionnaire results that we obtained here and based on [18] is given in Figure 9. We used the Spearman-Brown formula to make all of the reliability estimates applicable to instruments of equal lengths. The two values chosen were 26 and 85 items, to match the full length instruments of the 1987 maturity questionnaire and the SPICE v1 capability dimension.

A point that can be noted from Figure 9 is that all of the internal consistency values for full length instruments are above the 0.9 minimal threshold. Therefore, both full length instruments can be considered internally consistent for practical purposes. Also, the SPICE capability dimension tends to be slightly more internally consistent than the 1987 maturity questionnaire, whichever instrument length is chosen.

From a practical perspective it would not be reasonable to reduce the maturity questionnaire to 26 items because that would negatively impact its validity (i.e., the extent to which it is measuring organizational maturity). Similarly, it would not be practical to increase the SPICE capability dimension to 85 items because that would dramatically increase the cost of conducting assessments. However, it is of practical interest to determine the features of the SPICE v1 capability dimension that makes it more intrinsically reliable. Such knowledge can help us improve future assessment instruments.

5. Conclusions

In this paper we evaluated the internal consistency of the 1987 SEI maturity questionnaire and the v1 SPICE capability dimension. We found that the two assessment instruments as they are constructed have very high internal consistency values. The SPICE capability dimension, however, tends to have slightly larger values. We found the results to be consistent across two samples for each instrument. This gives us some initial confidence in the usage of these instruments.

Further research ought to focus on three issues. The first is that it would be useful for future assessment instrument construction to determine the reasons behind the larger SPICE capability dimension internal consistency values. Second, estimation of other types of reliability using other methods is necessary, for example, inter-rater agreement coefficients. Different types of reliability estimation take into account different sources of error. While there have been some studies of the interrater agreement in SPICE-based assessments (e.g., [12][15]), no published reports of interrater agreement studies of SEI assessment methods are known to the authors. Finally, now that we have established a baseline, it would be prudent to compare the internal consistency of the new version of the maturity questionnaire and the new version of the SPICE capability dimension to these baselines. It is then that we will know whether, at least in terms of internal consistency, the new instruments are an improvement over the previous versions.

Acknowledgments

The authors wish to thank Sasa Dekleva for providing us with his maturity data. We also wish to thank the assessors and sponsoring organizations who took part in the SPICE assessments. The authors also thank Sasa Dekleva and Dennis Goldenson for their comments on an earlier version of this paper.

6. References

[1] M. Allen and W. Yen: *Introduction to Measurement Theory*. Brooks/Cole Publishing Company, 1979.

[2] T. Bollinger and C. McGowan: “A Critical Look at Software Capability Evaluations”. In *IEEE Software*, pages 25-41, July 1991.

[3] E. G. Carmines and R. A. Zeller: *Reliability and Validity Assessment*, Sage Publications, Beverly Hills, 1979.

[4] L. Cronbach: “Coefficient Alpha and the Internal Structure of Tests”. In *Psychomterika*, 16(3):297-334, 1951.

[5] B. Curtis: “The Factor Structure of the CMM and Other Latent Issues”. Paper presented at the *Empirical Studies of Programmers: Sixth Workshop*, Washington DC, 1996.

[6] S. Dekleva: “Delphi Study of Software Maintenance Problems”, In *Proceedings of the International Conference on Software Maintenance*, pages 10-17, 1992.

[7] D. Drehmer and S. Dekleva: “Measuring Software Engineering Maturity: A Rasch Calibration”. In *Proceedings of the International Conference on Information Systems*, pages 191-202, 1993

[8] D. Dunaway and S. Masters: *CMM-Based Appraisal for Internal Process Improvement (CBA IPI): Method Description*. Technical Report CMU/SEI-96-TR-7, Software Engineering Institute, 1996.

[9] K. El Emam and D. R. Goldenson: “SPICE: An empiricist's perspective”. In *Proceedings of the Second IEEE International Software Engineering Standards Symposium*, pages 84-97, August 1995.

[10] K. El Emam and N. H. Madhavji: “The Reliability of Measuring Organizational Maturity”. In *Software Process Improvement and Practice*, 1(1):3-25, September 1995.

[11] K. El Emam and D. R Goldenson: “An Empirical Evaluation of the Prospective International SPICE Standard”. In *Software Process Improvement and Practice*, 2(2):123-148, 1996.

[12] K. El Emam, D. R. Goldenson, L. Briand, and P. Marshall: “Interrater Agreement in SPICE-based Assessments: Some Preliminary Results”. In *Proceedings of the Fourth International Conference on the Software Process*, pages 149-156, 1996.

[13] K. El Emam and N. H. Madhavji: “Does Organizational Maturity Improve Quality?”. In *IEEE Software*, pages 109-110, September 1996.

[14] K. El Emam and D. R. Goldenson: "Description and evaluation of the SPICE Phase one trials assessments". In *Proceedings of the International Software Consulting Network Conference* (ISCN'96), 1996.

[15] K. El Emam, L. Briand, and R. Smith: “Assessor Agreement in Rating SPICE Processes”. To appear in *Software Process Improvement and Practice*, 1997.

[16] T. Olson, W. Humphrey, and D. Kitson: *Conducting SEI-Assisted Software Process Assessments*. Technical Report CMU/SEI-89-TR-7, Software Engineering Institute, 1989.

[17] W. Humphrey and W. Sweet: *A Method for Assessing the Software Engineering Capability of Contractors*. Technical Report CMU/SEI-87-TR-23, Software Engineering Institute, 1987.

[18] W. Humphrey and B. Curtis: “Comments on a ‘A Critical Look’”. In *IEEE Software*, pages 42-46, July 1991.

[19] W. Humphrey, D. Kitson, and J. Gale: “A Comparison of US and Japanese Software Process Maturity”. In *Proceedings of the 13th International Conference on Software Engineering*, pages 38-49, 1991.

[20] ISO/IEC: *Software Process Assessment - Part 2: A Model for Process Management*, Working Draft 1.00, 1995.

[21] ISO/IEC: *Software Process Assessment - Part 4: Guide to Conducting Assessments*, Working Draft 1.00, 1995.

[22] P. Lamal: “On the Importance of Replication”. In *Replication Research in the Social Sciences*, J. Neuliep (ed.), Sage Publications, 1991.

[23] F. Maclennan and G. Ostrolenk: “The SPICE Trials: Validating the Framework”. In *Proceedings of the 2nd International SPICE Symposium*, Brisbane, Australia, 1995.

[24] J. C. Nunnally: *Psychometric Theory*. McGraw-Hill, New York, 1967.

[25] M. Paulk and M. Konrad: “Measuring Process Capability versus Organizational Maturity”. In *Proceedings of the 4th International Conference on Software Quality*, 1994.

[26] T. Rout: “SPICE: A Framework for Software Process Assessment”. In *Software Process Improvement and Practice*, Pilot Issue, pages 57-66, 1995.

[27] Members of the CMM-Based Appraisal Project: *Software Capability Evaluation (SCE) version 2.0: Implementation Guide*. Technical Report CMU/SEI-94-TR-5, Software Engineering Institute, 1994.

[28] D. Rugg: “Using a Capability Evaluation to Select a Contractor”. In *IEEE Software*, pages 36-45, July 1993.

[29] SEI: “Process Maturity Profile of the Software Community: 1996 Update”. Software Engineering Institute, 1996.

[30] A. Topper and P. Jorgensen: “More Than One Way to Measure Process Maturity”. In *IEEE Software*, pages 9-10, .

[31] M. Tan and C. Yap: “Impact of Organizational Maturity on Software Quality”. In *Software Quality and Productivity: Theory, Practice, Education and Training*, M. Lee, B-Z Barta, and P. Juliff (eds.), Chapman & Hall, 1995.

[32] R. Whitney, E. Nawrocki, W. Hayes, and J. Siegel: *Interim Profile: Development and Trial of a Method to Rapidly Measure Software Engineering Maturity Status*. Technical Report CMU/SEI-94-TR-4, Software Engineering Institute, 1994.

[33] D. Zubrow, W. Hayes, J. Siegel, and D. Goldenson: *Maturity Questionnaire*. Technical Report CMU/SEI-94-SR-7, Software Engineering Institute, 1994.

7. Appendix A

The SPICE architecture is two dimensional[4]. Each dimension represents a different perspective on software process management. One dimension consists of *processes*. Each process contains a number of *base practices*. A base practice is defined as a software engineering or management activity that addresses the purpose of a particular process. Processes are grouped into *Process Categories*. An example of a process is *Develop System Requirements and Design*. Base practices that belong to this process include: *Specify System Requirements*, *Describe System Architecture*, and *Determine Release Strategy*. An overview of the process categories is given in Figure 10.

The other dimension consists of *generic practices*. A generic practice is an implementation or institutionalization practice that enhances the capability to perform a process. Generic practices are grouped into *Common Features*, which in turn are grouped into *Capability Levels*. An example of a Common Feature is *Disciplined Performance*. A generic practice that belongs to this Common Feature stipulates that data on performance of the process must be recorded. An overview of the Capability Levels is given in Figure 11.

Ratings are made against the capability dimension. Each generic practice is rated based on its implementation in the process. This rating utilizes a four-point adequacy scale. The four discrete values are summarized in Figure 12. The four values are also designated as F, L, P, and N.

Process Category	Description
Customer-supplier	processes that directly impact the customer, supporting development and transition of the software to the customer, and provide for its correct operation and use
Engineering	processes that directly specify, implement or maintain a system and software product and its user documentation
Project	processes which establish the project, and co-ordinate and manage its resources to produce a product or provide services which satisfy the customer
Support	processes which enable and support the performance of the other processes on a project
Organization	processes which establish the business goals of the organization and develop process, product and resource assets which will help the organization achieve its business goals

Figure 10: Brief description of the Process Categories.

[4] Elements of the SPICE architecture have recently been revised and restructured. The basic two dimensional architecture remains however. In this study, we used the first version of the SPICE documents only.

Capability Level	Description
Level 0: Not Performed	There is general failure to perform the base practices in the process. There are no easily identifiable work products or outputs of the process.
Level 1: Performed-Informally	Base practices of the process are generally performed, but are not rigorously planned and tracked. Performance depends on individual knowledge and effort. There are identifiable work products for the process.
Level 2: Planned-and-Tracked	Performance of the base practices in the process is planned and tracked. Performance according to specified procedures is verified. Work products conform to specified standards and requirements.
Level 3: Well-Defined	Base practices are performed according to a well-defined process using approved, tailored versions of the standard, documented processes.
Level 4: Quantitatively-Controlled	Detailed measures of performance are collected and analyzed leading to a quantitative understanding of process capability and an improved ability to predict performance. Performance is objectively managed. The quality of work products is quantitatively known.
Level 5: Continuously-Improving	Quantitative process effectiveness and efficiency goals for performance are established, based on the business goals of the organization. Continuous process improvement against these goals is enabled by quantitative feedback.

Figure 11: Brief description of the capability levels.

Rating & Designation	Description
Not Adequate - N	The generic practice is either not implemented or does not to any degree satisfy its purpose
Partially Adequate - P	The implemented generic practice does little to contribute to satisfy the purpose
Largely Adequate - L	The implemented generic practice largely satisfies its purpose
Fully Adequate - F	The implemented generic practice fully satisfies its purpose

Figure 12: Description of the rating scheme for generic practices.

8. Appendix B

This appendix contains the exact text of the questions that were used to evaluate the 1987 SEI maturity questionnaire. Respondents answered Yes or No to each question. We indicate in the tables whether a question was included in the first and second data sets. The numbers in the first column reference the question numbers in [17].

Maturity related questions - Level 2		Data set 1	Data set 2
1.1.3	Does the Software Quality Assurance (SQA) function have a management reporting channel separate from the software development project management ?	x	x
1.1.6	Is there a software configuration control function for each project that involves software development ?	x	x
2.1.13	Is a formal procedure used in the management review of each significant software development prior to making contractual commitments ?	x	x
2.1.14	Is a formal procedure used to make estimates of software size ?	x	x
2.1.15	Is a formal procedure used to produce software development schedules ?	x	x
2.1.16	Are formal procedures applied to estimating software development cost ?	x	x
2.2.2	Are profiles of software size maintained for each software configuration item, over time ?	x	x
2.2.4	Are statistics on software code and test errors gathered ?	x	x
2.4.1	Does senior management have a mechanism for the regular review of the status of software developments projects ?	x	x
2.4.7	Do software development first-line managers sign off on their schedules and cost estimates ?	x	x
2.4.9	Is a mechanism used for controlling changes to the software requirements ?	x	x
2.4.17	Is a mechanism used for controlling changes to the code ? (Who can make changes and under which circumstances ?)	x	x
2.1.7	For each project, are independent audits conducted for each step of the software development process ?		x
2.2.1	Are software staffing profiles maintained of actual staffing versus planned staffing ?		x
1.2.2	Is there a required training program for all newly appointed development managers designed to familiarize them with software project management ?		x
2.2.16	Are software trouble reports resulting from testing tracked to closure ?		x
2.2.8	Are profiles maintained of actual versus planned software units completing unit testing, over time ?		x
2.1.4	Is a formal procedure used to assure periodic management review of the status of each software development project ?		x
2.1.17	Is a mechanism used for ensuring that the software design teams understand each software requirement ?		x
2.4.5	Is a mechanism used for regular technical interchanges with the customer ?		x
2.2.9	Are profiles maintained of actual versus planned software units designed, over time ?		x

	Maturity related questions - Level 3	Data set 1	Data set 2
1.1.7	Is there a software engineering process group function ?	x	x
1.2.3	Is there a required software engineering training program for software developers ?	x	x
1.2.5	Is a formal training program required for design and code reviews leaders ?	x	x
2.1.1	Does the software organization use a standardized and documented software development process on each project ?	x	x
2.2.3	Are statistics on software design errors gathered ?	x	x
2.2.15	Are the action items resulting from design reviews tracked to closure ?	x	x
2.2.17	Are the action items resulting from code reviews tracked to closure ?	x	x
2.4.6	Is a mechanism used for ensuring compliance with the software engineering standards ?	x	x
2.4.12	Are internal software design reviews conducted ?	x	x
2.4.13	Is a mechanism used for controlling changes to the software design ?	x	x
2.4.16	Are software code reviews conducted ?	x	x
2.4.19	Is a mechanism used for verifying that the samples examined by Software Quality Assurance are truly representative of the work performed ?	x	x
2.4.21	Is there a mechanism for assuring the adequacy of regression testing ?	x	x
2.4.8	Is a mechanism used for ensuring traceability between the software requirements and top-level design ?		x
2.1.10	Are standards applied to the preparation of unit test cases ?		x
1.2.4	Is there a required software engineering training program for first-line supervisor of software development ?		x
2.4.18	Is a mechanism used for software configuration management ?		x
2.1.12	Does the standard software development process documentation describe the use of tools and techniques ?		x
2.1.6	Are standards used for the content of software development files / folders ?		x
2.1.8	Is a mechanism used for assessing existing designs and code for reuse in new applications ?		x
1.3.2	Is a mechanism used for evaluating technologies used by the organization versus those externally available ?		x
2.4.11	Is a mechanism used for ensuring traceability between the software top-level and detailed design ?		x
2.4.22	Are formal test cases review conducted ?		x

Maturity related questions - Level 4		Data set 1	Data set 2
2.3.1	Has a managed and controlled process database been established for process metrics data across all projects ?		x
1.3.4	Is a mechanism used for managing and supporting the introduction of new technologies ?		x
2.2.14	Is test coverage measured and recorded for each phase of functional testing ?		x
2.3.8	Is review efficiency analyzed for each project		x
2.3.2	Are the review data gathered during design reviews analyzed ?		x
2.2.6	Are code and test errors projected and compared to actual ?		x
2.3.4	Are analyses of errors conducted to determine their process related causes ?		x
2.1.13	Are code reviews standards applied ?		x
2.4.2	Is a mechanism used for periodically assessing the software engineering process and implementing indicated improvements ?		x
2.2.13	Are design and code review coverages measured and recorded ?		x
2.3.3	Is the error data from code reviews and tests analyzed to determine the likely distribution and characteristics of the errors remaining in the product ?		x
2.2.5	Are design errors projected and compared to actual ?		x