On comparing process assessment results: BOOTSTRAP and CMM

J. Järvinen

Department of Information Processing Science, University of Oulu, Linnanmaa, FIN-90570 Oulu, Finland

ABSTRACT

Software process assessment approaches, models, methods and methodologies have gained much attention in both MIS and SE recently. Perhaps the most well-known approach is CMM (Capability Maturity Model) which has been developed by SEI (Software Engineering Institute at Carnegie Mellon University, Pittsburgh). One of the more well-known European assessment approaches is called BOOTSTRAP. The Bootstrap approach includes both the CMM and the ISO 9000 standard as its starting point. These Bootstrap origins are explained and discussed.

The Bootstrap and the SEI maturity questionnaires are compared using the Conte et al's [8] meta-metrics as criteria. Comparability of Bootstrap assessment results with SEI results has been examined using the empirical material collected from the software process assessments made in Europe during 1991-1993 under auspices of the ESPRIT/Bootstrap project. Finally, the future of software process assessment approaches is considered.

The conclusion is that the two questionnaires and algorithms are not directly comparable at the moment, but the Bootstrap method with its more holistic approach is more suitable than the SEI approach for assessing software producing units that are not subcontractors of the DoD.

INTRODUCTION

Shortcomings of the SEI model and the lack of European perspective have motivated the development of the Bootstrap assessment method.
Bootstrap has extended the SEI approach and adjusted it to suit European software production. This means primarily expanding the target group, adapting the ISO 9000 standards approach and adding state-of-the-art software engineering ideas to the model.

The Bootstrap consortium has always emphasized that their method retains comparability with the CMM approach. This paper investigates this claim and the enhancements of Bootstrap. An earlier version of this work was presented at IRIS-16 [17].

First, the CMM and Bootstrap software process assessment approaches are introduced and a brief comparison of the two questionnaires and algorithms is done. Secondly, the empirical Bootstrap assessment data is presented with the mapping of the Bootstrap questions to their SEI equivalents. Then the calculated CMM maturity results from Bootstrap assessment data are compared to the Bootstrap results and discussed. Lastly, the future of software process assessment approaches is considered.

SEI

In November 1986, the Software Engineering Institute (SEI), with assistance from the MITRE corporation, began developing a process maturity framework that would assist organizations in improving their software process. This effort was initiated in response to a request to provide the U.S. federal government with a method for assessing the capability of its software contractors.


In August 1991, the fully defined Capability Maturity Model (CMM) v. 1.0 was released and the current version 1.1 of the CMM was released in July 1993 as two technical reports (Paulk et al [20], [21]). The CMM continues to evolve but "SEI anticipates that CMM v. 1.1 will remain the baseline until at least 1996" (cf. Paulk et al [19]).

The CMM is used extensively and it has been established perhaps as the leading software process assessment method and model in the US. The SEI has also been criticized for flaws in the maturity model concepts and assessment methods (cf. e.g. Bollinger and McGowan [4], Card [6], Pressman [22]).
The five levels of software process maturity
A maturity level is a well defined plateau on the path toward becoming a mature organization. Each maturity level provides a layer in the foundation for continuous process improvement. Achieving each level of the maturity framework establishes a different component in the software process, resulting in an increase in the process capability of the organization (cf. Humphrey [13], Paulk et al [20]). The five levels of the maturity framework are shown in figure 1.

Figure 1: Process Maturity Levels (Humphrey [13]).

Paulk et al. [20] characterize the five maturity levels with the primary process changes made at each level as follows:

1) Initial The software process is characterized as ad hoc, and occasionally even chaotic. Few processes are defined, and success depends on individual effort.

2) Repeatable Basic management processes are established to track cost, schedule, and functionality. The necessary process discipline is in place to repeat earlier successes on projects with similar applications.

3) Defined The software process for both management and engineering activities is documented, standardized, and integrated into an organization-wide software process. All projects use a documented and approved version of the organization's process for developing and maintaining software.
4) Managed Detailed measures of the software process and product quality are collected. Both the software process and products are quantitatively understood and controlled using detailed measures.

5) Optimizing Continuous process improvement is enabled by quantitative feedback from the process and from testing innovative ideas and technologies.

BOOTSTRAP

Bootstrap was an ESPRIT technology transfer project whose one purpose was to develop a European method for assessing companies using the SEI software process maturity framework and questionnaire as the reference. However, it was felt that the SEI concentrated mainly on very large embedded defence systems so additional sources were needed.

The aim was to define an improvement path for software producing units with distinct milestones. The software process maturity framework by Humphrey [13] has provided a basis for the milestones and ISO 9001 [15] and ISO 9000-3 [16] have been used as a framework for the quality system in general.

The goal of the Bootstrap methodology is to help European software producing units (SPU) and companies to reduce risks relating to their products and projects. To achieve this goal management is the key factor. Technologies are marginal in respect to the growth of productivity and quality - the elements needed to reduce risk.

"In line with state of the art thinking and the better practices we are strongly convinced that three dimensions rule productivity and quality in software production:

- Organization
- Methodology
- Technology

These three dimensions are listed in order of relevance and consequentiality; only after the setting up of a proper organization is it relevant to apply methodologies. Only after that it becomes significant to support them by means of tools." (cf. Cachia [5]).
Bootstrap Questionnaire
The Bootstrap questionnaire version 1.0 was produced in the first phase of Bootstrap project. The questionnaire was derived from the SEI maturity questionnaire after an exercise of restructuring and porting of contents according to ISO-9000-3 form (see figure 2).

Version 2.22 was derived from version 1.0 as follows [5]:

• existing questionnaires were considered in order to attain increased coverage by including further sub-checklists and questions
• the SEI CMM was taken into account
• assessor experience and feedback from version 1.0 were carefully considered.

Other sources have also influenced the version 2.22 (cf. e.g. Rocky Mountain [23], ESA standards [11]).

The Bootstrap questionnaire is organized as a tree structure that identifies the main attributes to be analyzed. Figure 3 shows that the Bootstrap questionnaire consists of clusters of questions. Each question, a sensor, tests out one aspect or attribute of process. Koch [18] explains that Bootstrap is: "structurally comparable to Basili's

Figure 3: The Bootstrap questionnaire v. 2.22 structure.

Two different questionnaires are used in a Bootstrap assessment, a general questionnaire for the software producing unit (SPU) and another questionnaire for projects. Both questionnaires have basically the same question (sensor) set, with one important difference. An SPU question typically seeks to determine the existence of a certain attribute in the system, e.g. the existence of coding standards, whereas a project question typically seeks to determine whether that attribute has also been adopted in the project at hand.

Using the Bootstrap algorithm for calculating the maturity level for a software producing unit or project an attribute profile is generated. In the attribute profile the weaknesses and strengths of the assessed unit are displayed (see figure 4).
Maturity Level

Figure 4: An example of an SPU maturity profile.

Bootstrap has tried to ascertain that the questionnaire sufficiently covers the SE activity in projects and SPU's by using a mix of sources when designing the questionnaire. According to Koch [18] and Bicego et al [3], the response from the assessed has been very positive, so it seems that the Bootstrap approach has been satisfactory so far.

COMPARISON OF CMM AND BOOTSTRAP

The CMM and the Bootstrap maturity questionnaire with their respective algorithms are metrics used to quantitatively assess software processes. Therefore, it is possible to analyze them using general criteria for tentative metrics. In this chapter the two questionnaires and algorithms are examined using the Conte et al criteria for a metric. Also the general features of CMM and Bootstrap have been used in the comparison whenever possible.

Conte et al: Meta-metrics

Conte, Dunsmore & Chen [8] define meta-metrics, i.e. evaluation criteria for a metric. These meta-metrics are:

Simplicity

Does the metric lead to a simple result that is easily interpreted? A single, intuitive value - like the number of reported errors as a software quality metric - is
simple. However, another metric that provides several numbers or values - like logarithms or other metrics - which are not as easily interpreted is not simple.

**Validity**

Does the metric measure what it purports to measure? It is easy to see that the lines of code count is a valid measure for program size. But a measure for program comprehension may actually be the score on some multiple-choice test, whose value is, at best, indirectly related to comprehension.

**Robustness**

Is the metric sensitive to the artificial manipulation of some factors that do not affect the performance of the software? For example, the value of a metric should not be drastically changed by the rearrangement of code, which does not affect the execution of the program.

**Prescriptiveness**

Can the metric be used to guide the management of software development or maintenance? For example, in order to guide the development process, it is better if the value of the metric can be assessed during the development process, rather than only at the end. In this case, the manager may know when a problem occurs, and may take proper corrective action before it is too late.

**Analyzability**

Can the value of the metric be analyzed using standard statistical tools? For example, the common metric "lines of code" is easily analyzable, while the binary metric "use/nonuse of structured programming techniques" is not.

### Comparison of CMM and Bootstrap based on meta-metrics

Table 1 shows an evaluation of CMM and Bootstrap using the Conte et al criteria for a metric. The comparison is relative to the two methods and is made with the knowledge gathered during the Bootstrap project. The main arguments for the ratings are as follows:

**Simplicity** Since the software maturity framework defines maturity levels precisely, both CMM and Bootstrap produce a simple result; a single number that gives a distinct view of the assessed organization. Bootstrap also enables a closer look of an SPU and its projects with maturity profiles of the modelled attributes. A similar construct, the key process area profile, has been added to CMM.

**Validity** The two questionnaires are composite metrics and partially subjective by nature. SEI, with only 85 graded questions, has also the problem of sparse data points (cf. Bollinger and McGowan [4]), which
Managing Quality Systems  255

can lead to inadequate coverage. Carleton [7] of the SEI admits this by saying that the questionnaire does not cover all key indicators (of CMM). Bootstrap's special problem with validity concerns the potential error possibilities with extended answer scale and the mixing of level-specific questions in maturity calculation. On the other hand, in both approaches the questions are very specific and act as sensors for properties defined in the maturity model. Future versions of Bootstrap questionnaire will strive for greater validity and objectivity with more explicit metrics and more context-sensitive questions.

Robustness The SEI maturity level calculation is greatly affected if some questions, specifically key questions, are omitted or answered incorrectly. We have observed this during the attempt to map Bootstrap questions to SEI. This is caused mainly by the multi-hurdle grading scheme, which Bollinger and McGowan [4] have discussed in detail. In CMM version 1.1 the maturity questionnaire does not have an important role (cf. Paulk et al [21]).

Bootstrap seems to be a very robust metric. This is achieved in two ways. Firstly, all questions are treated basically equally so omitting a question does not affect performance very much especially since omissions are allowed (n/a questions). Secondly, the maturity calculation is performed for each simple and composite attribute independently, which localizes the effects of possible errors.

Prescriptiveness Both methods are quite prescriptive because they are based on the capability maturity model, which defines the properties of individual maturity levels clearly. Bootstrap is very good in defining the improvement areas because each attribute; a cluster of questions, can be evaluated and compared to the overall score of the SPU or project. With CMM, the necessary activities for improvement are meticulously documented in the Key Practices of the CMM (cf. Paulk et al [21]).

Analyzability The SEI maturity questionnaire being a binary metric and using an ordinal maturity scale is not very easy to analyze. As CMM does not include any better metrics or consistent criteria (cf. Card [6]) for formal examination of assessment results, analyzing the results is even more difficult. Bootstrap is better with its extended answer and interval maturity scale, but is not without analyzability problems. The CMM and Bootstrap model complex operations in a complex situation resulting in somewhat unique data that cannot easily be analyzed statistically or compared without proper knowledge of the context. The improvement in analyzability is clearly one of the goals in the further development of Bootstrap.
Table 1. Evaluation of CMM and Bootstrap by meta-metrics criteria (Conte et al [8]).

### Summary
Using Conte et al's meta-metric criteria for evaluating the CMM and Bootstrap questionnaire and algorithm the conclusions are quite straightforward. Bootstrap has mainly advantages over the SEI approach so it seems to be a better metric, although the maturity calculation might produce some problems due to blending of level-specific questions. Finally, in order to improve, Bootstrap should take some issues into account from the current CMM to bring the Bootstrap assessment package to completion.

### AN EMPIRICAL STUDY ON COMPARABILITY

An analysis was done to study the comparability and differences between the CMM and Bootstrap approaches using sample data from assessments made during the Bootstrap project. Using the same empirical data many potentially disturbing variables are eliminated when examining the questionnaires and algorithms.

Parts of the BootTool – Bootstrap assessment analyzer and database – software package were created during the study, such as the mapping of each version of Bootstrap questionnaire to the SEI maturity questionnaire. The BootTool software package now includes routines for SPU or project maturity level calculation by either Bootstrap or SEI algorithm.

### Hypothesis for the study
A leading thought in the development of the Bootstrap method and questionnaire has been to preserve comparability with the original SEI questionnaire. Obviously, comparable assessment results would give interesting insight on the capability status of U.S.A. and Europe. A hypothesis was that the SEI questionnaire is a subset of Bootstrap
questionnaire so a direct functional dependence is always found (see Figure 5). The SEI metric yields results \( S_1 - S_n \) and the Bootstrap metric results \( B_1 - B_n \). Are \( S_i = B_i \)? If so, what is the added value and rationale of Bootstrap. What factors there are if \( S_i \neq B_i \)?

![Figure 5: Proposed relationship between an SPU and the Bootstrap and SEI methods.](image)

**Bootstrap assessment data**
During the Bootstrap project there were Bootstrap assessments done in Finland, Italy, Germany, Austria, Belgium and Switzerland. A data sample of 127 assessments was used in this study.

**Mapping of Bootstrap questions to SEI questions**
The mapping of the Bootstrap questionnaires to the SEI maturity questionnaire was done for three reasons.

Firstly, Bootstrap wanted to find out whether the questionnaire had included all the equivalent questions from the SEI questionnaire and what questions were added.

The second reason was to produce a mapping to facilitate the automatic calculation of the SEI maturity level number using data from any Bootstrap assessment. This meant creating a separate mapping of the SPU questionnaire and the project questionnaire for each major Bootstrap questionnaire version (1.0, 2.1, 2.22). Thus, six different mappings were formed. This paper deals only with the most current questionnaire version (2.22) and its mapping.

The third reason was to detect any errors and inconsistencies in the Bootstrap questionnaires.

**Mapping results** The hypothesis concerning the direct functional dependence (see figure 5) was not correct. We have found inconsistencies within the Bootstrap questionnaire concerning comparability.
between the two questionnaires. Twenty-seven out of 102 questions (26.5%) were entirely left out from the Bootstrap questionnaire version 2.22. Most questions (11) were taken from the Process Control section. Of the 27 omissions, four questions were attributed as key questions by the SEI. Also some minor problems were encountered. These typically related to different scope or a conflicting level of a question.

While the Bootstrap questionnaire version 2.22 still needs to be augmented with the 27 missing questions from SEI questionnaire, Bootstrap has also extended the v. 2.22 questionnaire by 83 new questions. These questions add depth to the SPU and project assessments. Most of the new questions concern the software development life-cycle functions, which have been neglected by the SEI.

**Calculated SEI results**

The mapping between the two questionnaires enabled the calculation of maturity level number according to the SEI maturity calculation algorithm. The interest for this simulation was to examine how the SEI results compare with maturity level numbers calculated according to the Bootstrap algorithm. This calculation was performed using the BootTool - software package that was built by the Bootstrap project team.

The data from Bootstrap assessments was used to calculate the SEI maturity level number for each assessed unit. The calculation produced following results (see figure 6):

Companies and projects which are clearly at the Initial maturity level are usually detected well, but many of those units judged to be at the Repeatable level by Bootstrap stay at level 1 according to the SEI results simulation. Further, some of the units (3%) reach the Defined level by SEI. While a clear breach in comparability exists, the rigidness of the SEI questionnaire and algorithm must be noted. One possible reason for some assessed companies not to pass the first hurdle is the background of the CMM questionnaire. It was developed to assess DoD software subcontractors invoking DoD 2167A standard [9] in their projects that usually are large real-time (embedded) systems. The practices required in that environment are not likely to be repeated or even needed in some other circumstances.

The effect of missing questions was also checked with two scenarios where the missing Bootstrap questions were treated with a "yes" and a "no" answer in the SEI mapping. These experiments did not have a notable effect on the SEI maturity scores.
Some problems were caused by the different answer scaling. A 'No' answer in SEI can mean 'not applicable' or 'No' in Bootstrap. A more serious problem arises with the 'Yes' answers that have three levels of intensity (2,3,4) in Bootstrap. Thus, a company with many barely existent practices (2's) will have the same SEI result as a company with the same number of excellent practices (4's). The Bootstrap results for these two companies will vary a great deal.

![Comparison of calculated SEI and Bootstrap results](image)

Figure 6: Comparison of calculated SEI and Bootstrap results

**Summary**

The results show that the results calculated with the Bootstrap and the SEI algorithms do not correspond very well at the moment. This small empirical study strengthens the view that comparability between questionnaires and algorithms can only be achieved when the assessed units are by nature similar to those the questionnaires were designed for. Specifically, SEI was designed for assessing large DoD projects, and it lacks adequate coverage for many 'ordinary' projects or software producing units.

**DISCUSSION**

It is very difficult to compare the assessment results of software producing units and their processes. Ideally, the same result should be achieved using different sets of questions. This might be possible if both sets of questions were based on the same model or they both were capable of describing the real-life situation in a company correctly. While the latter is something to ultimately aim at, the former is still quite a challenge to do. Palk et al [20] describe a structure (see figure 7) with which different question sets could be designed. This is
also a proposal for a more general framework of improvement of software processes. Other initiatives where the uniformity of software process assessment approaches is desired are the ImproveIT project [1] and the SPICE effort [10].

![Diagram of the Capability Maturity Model](image)

Figure 7: The structure of the Capability Maturity Model (Paulk et al [20]).

Although the aim for more general frameworks is beneficial for the software industry there remains much work in uncovering the specific properties of the different types of software producers. An organization which does telecommunications software is very different from an accounting software company; a real-time process control software company has other requirements for its organization and processes than an in-house MIS unit.

Further, in designing a good improvement path we need means for obtaining additional coverage of the complex world of software pro-
duction as the software process is only a part of it. Other approaches could survey the job satisfaction of employees, the perceived quality and performance of the supplier as seen by its customers or financial success of the organization. After all, as Pressman [22] says: "the primary goal isn't a level 5 on the SEI scale. It's developing high-quality software that's delivered on time and within budget."

REFERENCES


