CMM®-Based Process Improvement and Schedule Deviation in Software Maintenance

Ho-Won Jung
Dennis R. Goldenson

July 2003

Software Engineering Measurement and Analysis Initiative

Technical Note
CMU/SEI-2003-TN-015

Contents

List of Figures

List of Tables

Acknowledgements

Thanks are due first and foremost to the assessors, sponsors, and others who participate in assessments of the Capability Maturity Model® (CMM®) for Software (SW-CMM). This work would not be possible without the information that they regularly provide to the Software Engineering Institute (SEI℠).

Mike Zuccher, Kenny Smith, and Xiaobo Zhou provided invaluable support in extracting the data on which the study is based. Special thanks go to Sheila Rosenthal for her expert support with our references, and to Lauren Heinz for helping to improve the readability of the document. We also express our thanks to our SEI colleagues—Will Hayes, Mike Konrad, Steve Masters, Jim McCurley, Mark Paulk, Mike Philips, Gene Miluk, and Dave Zubrow—for their valuable comments on this work.

® Capability Maturity Model and CMM are registered in the U.S. Patent and Trademark Office by Carnegie Mellon University.

℠ SEI is a service mark of Carnegie Mellon University.

Abstract

The objective of this study is to evaluate the predictive validity of the Capability Maturity Model® (CMM®) for Software (SW-CMM) as applied to software maintenance.

The SW-CMM is intended to apply to both software development and maintenance. A basic premise (hypothesis) of the SW-CMM is that improving process maturity will result in better project performance and product quality. The extent to which that hypothesis is supported empirically is called a test of its predictive validity. No previous evaluation exists of the predictive validity of the SW-CMM in a maintenance context.

The extent to which schedule estimates differ from reality is one important measure of project performance. But is higher maturity in fact correlated with a reduction in schedule deviation? Data from 752 maintenance projects drawn from 441 SW-CMM assessments are analyzed using a zero inflated Poisson (ZIP) regression model, and the results are validated using a bootstrap estimation method. Projects from higher maturity organizations typically report less schedule deviation than those from organizations assessed at lower maturity levels.

® Capability Maturity Model and CMM are registered in the U.S. Patent and Trademark Office by Carnegie Mellon University.

1 Introduction

1.1 The Importance of Software Maintenance

The Capability Maturity Model® (CMM®) for Software (SW-CMM) [Paulk et al. 93a-93c, Paulk et al. 96] cites the definition of *maintenance* from IEEE Std 610-1990 [IEEE 90] as "the process of modifying a software system or component after delivery to correct faults, improve performance or other attributes, or adapt to a changed environment." This definition includes at least three types of software maintenance:

1. **corrective maintenance**: To correct processing, performance, or implementation faults of the software.
2. **adaptive maintenance**: To adapt the software to changes in environment such as new hardware of the next release of an operating system. Adaptive maintenance does not lead to changes in the system's functionality.
3. **perfective maintenance**: To perfect the software for its performance, processing efficiency, maintainability, or accommodation of new or changed user requirements.

The IEEE has estimated the annual cost of software maintenance in the United States to exceed $70 billon [Edelstein 93, Lerner 94]. Schrank has estimated it to be more than $30 billion annually [Schrank et al. 95]. Others have estimated the magnitude of software maintenance costs to range from 40 to 80 percent of overall software life-cycle costs [Alkhatib 92, Kemerer 95, Schrank et al. 95]. A widely used rule of thumb for the distribution of maintenance activities has been 60 percent for enhancements, 20 percent for adaptation, and 20 percent for error correction [Lientz & Swanson 80, Glass & Noiseux 81].

While the SW-CMM is intended to be suited for both development and maintenance processes, difficulties in implementing the model in maintenance-only organizations have been reported [Drew 92]. Others have criticized the SW-CMM for not directly addressing maintenance [Kuilboer & Ashrafi 00]. One survey study conducted in the United Kingdom failed to find evidence that higher maturity companies manage maintenance more effectively than lower maturity companies; however, the survey does not explicitly state how it defines maturity [Hall et al. 01]. Swanson and Beath claimed that software maintenance is fundamentally different from development of new systems since the maintainer must interact with an existing system [Swanson & Beath 89].

® Capability Maturity Model and CMM are registered in the U.S. Patent and Trademark Office by Carnegie Mellon University.

Niessink and van Vliet investigated the difference between software maintenance and software development from a service point of view [Niessink & van Vliet 00]. They argued that software maintenance can be seen as providing a service, while software development is concerned with the development of products. Hence, they developed a separate information technology (IT) service Capability Maturity Model meant for software maintenance organizations and other IT service providers. Similarly, Kajko-Mattsson developed a problem management maturity model for corrective maintenance [Kajko-Mattsson 02].

1.2 This Study

A basic premise of the SW-CMM is that higher process maturity is associated with better project performance and product quality. Furthermore, improving maturity is expected to subsequently improve both performance and quality. Testing this premise can be considered an evaluation of the predictive validity of the assessment measurement procedure [El-Emam & Goldenson 95]. Given both the high cost of software maintenance and enduring questions about the applicability of the SW-CMM, it is important to provide objective evidence about the predictive validity of the SW-CMM in a maintenance context.

This study provides evidence that higher process maturity is in fact associated with "reduced mean and variance" of schedule deviation in software maintenance.[1] The analysis is based on 752 maintenance projects from 441 CMM-Based Appraisals for Internal Process Improvement (CBA IPI) assessments. A zero inflated Poisson (ZIP) regression model is used to account for nonnegative integer values and the existence of multiple reports of no deviations in schedule. The results are validated using a bootstrap estimation method.

Section 2 reviews previous studies on predictive validity and presents the study's hypotheses. Section 3 addresses data collection and the characteristics of our sample. Section 4 presents a brief introduction of a ZIP regression model and a bootstrap method for examining the stability of our results. Section 5 presents the results of the analysis. Section 6 contains our conclusions and final remarks.

[1] While the results are similar across the software development life cycle, important distinctions will be addressed in a subsequent study.

2 Empirical Hypotheses of Predictive Validity

2.1 Theoretical Basis

The SW-CMM provides a framework for organizing software processes into five evolutionary steps, or maturity levels, which lay successive foundations for continuous process improvement (Table 1). The SW-CMM covers practices for planning, engineering, and managing software development and maintenance. More mature software organizations, when following these key practices, are expected to be better able to meet their cost, schedule, functionality, product quality, and other performance objectives [Paulk et al. 96].

Table 1: Maturity Levels and their Key Process Areas [Paulk 99]

Level	Focus	Key Process Areas
Level 5 Optimizing	Continuous process improvement	- Defect Prevention - Technology Change Management - Process Change Management
Level 4 Managed	Product and process quality	- Quantitative Process Management - Software Quality Management
Level 3 Defined	Engineering processes and organizational support	- Organization Process Focus - Organization Process Definition - Training Program - Integrated Software Management - Software Product Engineering - Intergroup Coordination - Peer Review
Level 2 Repeatable	Project management processes	- Requirements Management - Software Project Planning - Software Project Tracking and Oversight - Software Subcontract Management - Software Quality Assurance - Software Configuration Management
Level 1 Initial	Competent people (and heroics)	

Testing the above basic premise of the SW-CMM requires an empirical evaluation of the predictive validity of the process maturity concept. Is there a characteristic relationship between process maturity and independently measured performance criteria? Clearly, such relationships may depend on other contextual factors; that is, the relationships may differ from one context to another or may exist in only a few contexts. This theoretical basis for evaluating predictive validity is depicted in Figure 1.

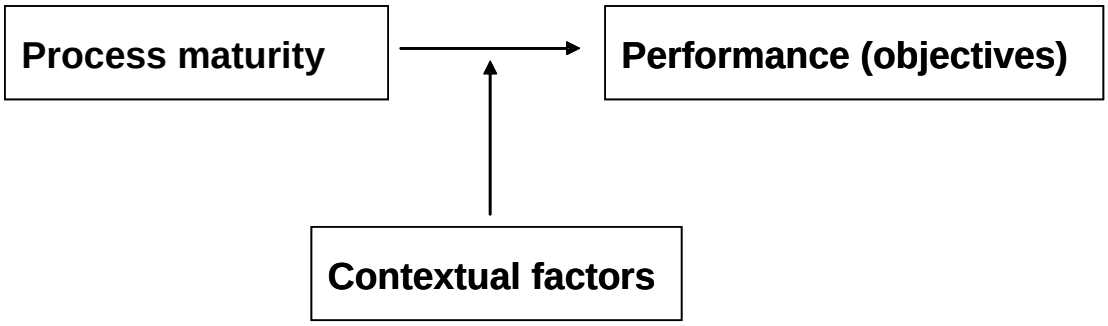

Figure 1: Theoretical Basis in a Predictive Validity Study

2.2 Variable Definition and Empirical Hypotheses

In the context of software maintenance in Figure 1, schedule deviation is the performance measure we use as our dependent variable. *Schedule deviation* is defined as the absolute value of the difference between actual schedule and planned schedule, i.e., y =|Actual-Planned|. Schedule deviation y is expressed in months ahead or behind schedule, with a value of zero indicating that the project is on schedule.[2]

Our explanatory variable, process maturity, is coded from maturity level 5 down to maturity level 1. Maturity level is an ordinal scale, not an interval scale; however, we do employ parametric statistics in this analysis.

Previous studies show that the distribution of maturity levels differs between the United States and elsewhere in the world [SEI 02, Jung et al. 02]. Hence, we examine how the region where the assessment was conducted (U.S. versus non-U.S.) acts as a contextual factor in mediating the effects of our research hypotheses.[3]

The theoretical basis shown in Figure 1 implies that schedule deviation is negatively associated with maturity level: the higher the maturity level, the less schedule deviation. In addition, the association may differ across regions of the world. Two types of benefits are expected to follow:

---

[2] One might argue that being ahead of schedule is less serious than being behind schedule; however, too few projects reported being ahead of schedule to allow a separate analysis here. Other weaknesses of the schedule deviation measure are described in Section 3.3.

[3] Classical measurement theory posits that variables should be measured on at least an interval scale to permit the computation of the mean and related parametric statistics [Stevens 51, Nunnally & Bernstein 94], but using only nonparametric methods on non-interval scale data would exclude much useful study [Nunnally & Bernstein 94]. Hence, many authors argue that a useful study can be conducted even if the proscriptions are violated [Briand et. al. 96, Gardner 75, Stevens 51, Velleman & Wilkinson 93]. El-Emam and Birk provide a detailed discussion of the scale type issue in studies of process capability and maturity [El-Emam & Birk 00a-00b].

- HYPOTHESIS 1: Increasing maturity level reduces the mean of schedule deviance in maintenance projects.
- HYPOTHESIS 2: Increasing maturity level reduces the variance of schedule deviance in maintenance projects.

Testing these two hypotheses in software maintenance projects allows us to evaluate the predictive validity of the process maturity concept. The same two hypotheses have been depicted elsewhere in graphical form as seen in Figure 2 [Paulk et al. 93a-93c, Paulk et al. 96].

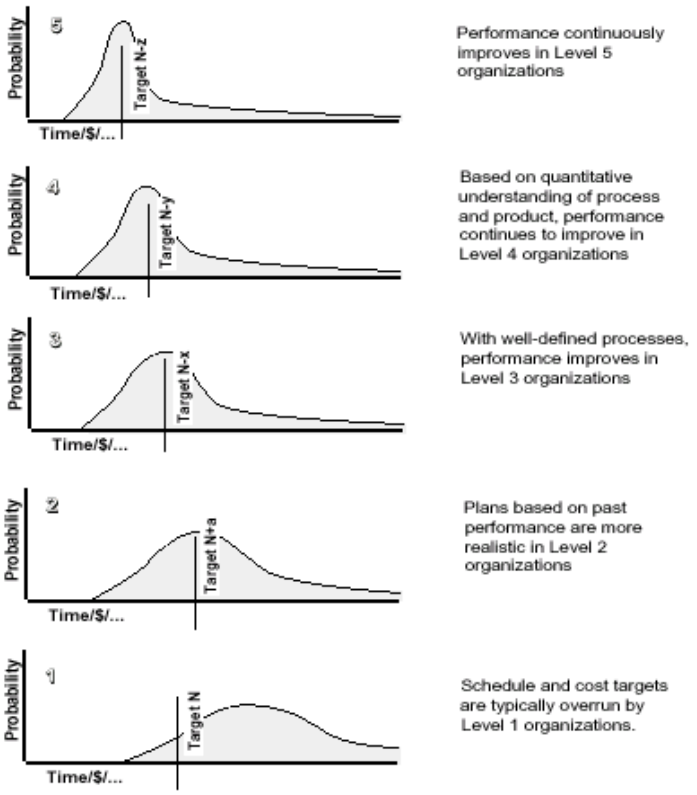

Figure 2: Process Capability as Indicated by Maturity Level [Paulk et al. 93a]

2.3 Previous Empirical Studies

All previous studies of predictive validity in process improvement are based either implicitly or explicitly on the theoretical model depicted in Figure 1. While some empirical studies examine variation across large numbers of organizations, most of them are case studies that describe the experiences and benefits from increasing process maturity in a single organization or a small number of organizations.

Case studies are quite useful for demonstrating proof of concept. There clearly are organizations that have benefited from increased process maturity [Brodman & Johnson 02, Butler 95, Diaz & Sligo 97, Daiz & King 02, Dion 92 & 93, Krasner 99].

Case studies, however, have a serious methodological disadvantage. It is difficult at best to generalize their results to a wider population. A case study can monitor projects in depth, but it is difficult to replicate the results later in a comparable context. Case studies also tend to suffer from a selection bias [adapted from El-Emam & Birk 00b]:

- Organizations that have not shown any process improvement or have even regressed will be highly unlikely to publicize their results, so case studies tend to show mainly success stories.
- The majority of organizations do not collect objective process and product data (e.g., on defect levels, or even keep accurate effort records). Only organizations that have made improvements and reached a reasonable level of maturity will have the actual objective data to demonstrate improvements (in productivity, quality, or return on investment). Therefore, failures and non-movers are less likely to be considered as viable case studies due to the lack of data.

By now, several predictive validity studies have collected data from larger numbers of organizations or projects, and they have statistically investigated relationships between capability maturity[4] and independent measures of performance. A survey study of individuals from SW-CMM-assessed organizations shows that higher maturity organizations tend to perform better on the subjective measures of performance (including ability to meet schedule), product quality, staff productivity, customer satisfaction, and staff morale [Goldenson & Herbsleb et al. 94, Herbsleb et al. 97]. In another survey-based study, Deephouse and colleagues found evidence of predictive validity in the relationships among seven software processes and measures of project performance including meeting schedule and budget targets, quality, and rework [Deephouse et al. 95, Deephouse et al. 96].

Lawlis and colleagues investigated the benefits of the SW-CMM with two measures extracted from U.S. Air Force contracts [Lawlis et al. 96]. Their results show that higher maturity projects typically perform better on indices of both cost and schedule performance than do those at a lower maturity level. In a study combining questionnaire data with existing project metrics, Krishnan and Kellner found that SW-CMM-based process maturity was associated characteristically with a reduction in delivered defects after correcting for size and personnel capability [Krishnan & Kellner 99].

El-Emam and Birk evaluated the predictive validity of the ISO/IEC 15504 (Software Process Assessment [ISO/IEC 96]) capability measure for four software processes: “Develop Software Requirements,” “Develop Software Design,” “Implement Software Design,” and “Integrate and Test Software” [El-Emam & Birk 00a, El-Emam & Birk 00b]. They found that the “develop software design” process was associated with several project performance measures. Using the same dataset, Hwang and Jung found that higher project-management process capability is related to increased productivity and improved morale in large

[4] Studies based on the SW-CMM have examined maturity level differences in performance. Studies based on ISO/IEC 15504 (Software Process Assessment [ISO/IEC 15504 96]), which uses a continuous representation, have examined differences in process capability.

organizations [Hwang & Jung 03]. However, much weaker relationships were found between project-management process capability and any of the performance measures in small organizations.

3 Data

3.1 Data Collection

3.1.1 Data Source

Authorized lead assessors are required to provide reports to the Software Engineering Institute (SEI[SM]) for their completed assessments. Assessment data on the reports are kept in an SEI repository called the Process Appraisal Information System (PAIS). The PAIS includes information for each assessment on the company and appraised entity, key process area (KPA) profiles, organization and project context, functional area representatives groups, findings, and related data.[5]

This report considers only CBA IPI assessments. Not all CBA IPI assessments include KPA rating profiles, since the determination of a maturity level or KPA ratings is optional and is provided at the discretion of the assessment sponsor. The dataset that we analyzed for this study was extracted from appraisal reports in the PAIS for the period of January 1998 through December 2001.

3.1.2 Dataset Analyzed

A statistical rule of thumb states that there should be at least six observations (sometimes five) to have confidence in analysis results. A similar criterion was used in an earlier analysis of software process assessment [Jung et al. 01]. Briand and colleagues [Briand et al. 00] and El-Emam and colleagues [El-Emam et al. 01] also have used a “greater-than-five-observations” criterion for the validation of software product metrics.

We follow the same rule of thumb here. Fewer than five maintenance projects at maturity levels 4 and 5 reported any schedule deviation whatsoever. Hence, we exclude maturity levels 4 and 5 from our statistical analysis. Note, however, that the lower incidence of reported schedule deviation at maturity levels 4 and 5 is of course entirely consistent with our empirical hypotheses.

[SM] SEI is a service mark of Carnegie Mellon University.

[5] Submitting an assessment report does not imply that the SEI certifies any assessment findings or maturity levels. All assessment data are kept confidential and are available only to SEI personnel on a need-to-know basis for research and development. Information in the PAIS is used to produce industry profiles or as aggregated data for research publications, and the SEI publishes a Process Maturity Profile twice a year (http://www.sei.cmu.edu/sema/profile.html).

Data exist for 752 maintenance projects from 441 organizations assessed at maturity levels 1 through 3 inclusive. Figure 3 shows the number of organizations and maintenance projects assessed by region. Since more than one maintenance project exists in some organizations, the number of organizations is fewer than the number of projects.

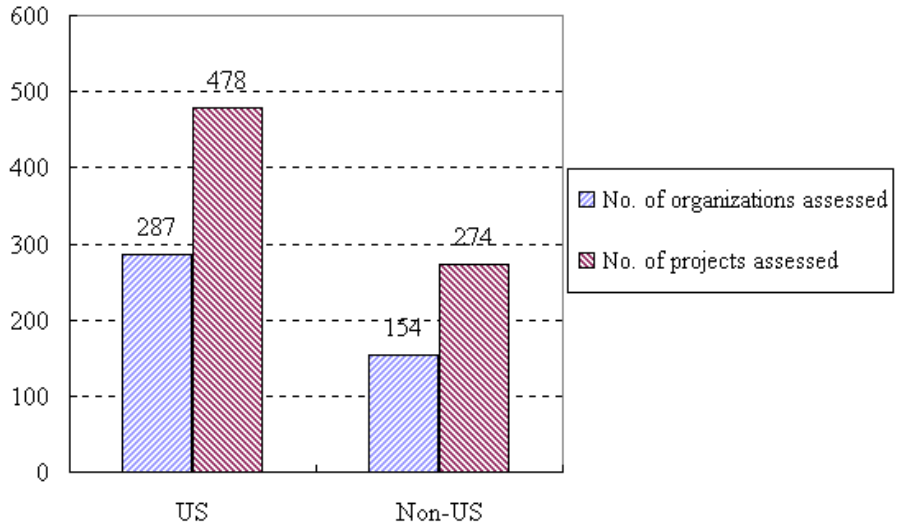

Figure 3: Organizations and Maintenance Projects in Regions

Table 2 shows the number of assessed maintenance projects at each maturity level. Schedule delays were reported by a total of 47 projects, while 8 projects reported being ahead of schedule.

Table 2: Number of Maintenance Projects

	Maturity Level 1	Maturity Level 2	Maturity Level 3	Total
U.S.	112 (12)	222 (6)	144 (5)	478 (23)
Non-U.S.	42 (6)	155 (20)	77 (6)	274 (32)

The numbers in parentheses denote the number of projects that reported deviations in schedule.

3.1.3 Unit of Analysis

The units of analysis in this study are projects in the maintenance phase of their life cycles, and our performance measure is schedule deviation expressed in months. Since the organization typically is the unit of analysis in CBA IPI assessments, our measure of maturity is organization-wide. If several maintenance projects are assessed in a single organization, all of the projects have the same level of maturity but have their own individual values of schedule deviation.

3.2 Data Quality

Our analysis mostly relies on two variables. The independent variable (covariate) is organizational maturity level as determined by CBA IPI assessment teams. A previous study provides ample confidence in the quality of that measure.[6] The dependent variable, schedule deviation, is a self-reported nonnegative integer measured by month, in which a project may be ahead, behind, or on schedule. Our reliance on such a measure raises significant accuracy issues.

In particular, a very large proportion, approximately 95 percent, of the projects in the maintenance phase of their life cycles reported being on schedule. That, of course, is contrary to both the results of previous studies and practical experience in the field.

Several reasons may account for this divergence. The question that is used to measure schedule deviation only asks whether or not the project is on time, but the criteria for being on time are not specified. One likely conjecture is that many projects periodically modify their baseline schedule estimates, which results in less-reported delay. Another is that assessments often include exemplary projects.

Time ahead or behind schedule is measured in months, so there also is most probably rounding error in the projects' replies. If a maintenance project is delayed for six weeks, should it be recorded as a delay of one month or two? Similarly, should a two-week delay be reported as a one-month delay or as essentially on time? Moreover, the measure does not account for variations in project size and duration. For example, a two-month delay in a one-month project is treated the same as a two-month delay in a nine-month project.

That said, as one might expect, reported schedule deviation is in fact higher for projects that are in other phases of their life cycles than maintenance. For example, more than 25 percent of the projects in test and integration do report being a month or more behind schedule.

Self-reports and direct observation often differ. For example, one study shows that software engineers over-report the amount of time that they work by an average of almost three percent; the proportion of times that self-reports and observer reports agreed on what the software engineer actually was doing varied substantially, from 95 to 58 percent [Perry et al. 96]. Errors in self-reports have been noted in various other studies, including voting [Abelson et al. 92], receiving of health care [Loftus et al. 92], and doctor's visits [McCallum et al. 95].

[6] In it we performed an internal-consistency reliability study using the same 676 CBA IPI assessments on which the present work is based [Jung and Goldenson 02]. The results identified three underlying dimensions of the capability maturity construct. "Project implementation" includes the key process areas (KPAs) at maturity level 2, "organization implementation" covers the KPAs at maturity level 3, and the KPAs at both maturity levels 4 and 5 are subsumed under "quantitative process implementation." Cronbach's alpha coefficient of internal consistency for each of the three dimensions exceeds the recommended value of 0.9, which indicates a sufficiently high level of internal consistency for use in practice.

Every measure has its strengths and weaknesses. For example, some studies recommend using relative measures.[7] But if the denominator has a small value, the measure may be exaggerated and take on an unreasonably large value.

Other candidate measures of schedule deviation include arithmetic means and standard deviations. However, they too are subject to a lack of robustness; one very small or very large value causes them to take on an arbitrarily large value. A trimmed method such as a Winsorized standard deviation or median absolute deviation might be used [Lunneborg 00]; however, such methods cannot be applied to a dataset characterized by excess zeros such as ours.

We have very little independent basis for judging the criterion validity of the schedule deviation question *per se*. Moreover, maturity level is an organizational construct while schedule deviation can vary by project. This too may introduce measurement error into our analysis. As will be seen later, however, our results are robust in spite of these limitations. The relationships with maturity level provide compelling evidence of the predictive validity of the SW-CMM.

3.3 Sampling Characteristics of the Dataset

Statistical analysis and its interpretations depend on the criteria by which a sample (subset) is selected from a population. Classical population inference requires random sampling. Hence, we examine here the sampling characteristics of our dataset.

The simplest form of sampling is a random sample. A *simple random sample* is defined as "a set of cases selected from a well-defined population of cases by a process that ensures that every sample containing the same number of cases has the same chance of being the one selected" [Lunneborg 00]. In the context of SW-CMM assessments, this definition explicitly requires two things: (1) a well-defined population of assessment cases from which to sample, and (2) a well-defined random process for selecting the sample.

The assessments reported to the PAIS database do not satisfy these two requirements. The population and the size of its assessments cannot be clearly defined, and the assessed organizations are not selected on a random basis. Rather, the assessments in PAIS are a self-selected sample (i.e., assessed organizations that have voluntarily participated in CBA IPI assessments to improve their software processes or were required to do so by contractors.) Our analyses here clearly must be based on nonrandom sampling methods.

[7] Conte and colleagues [Conte et al. 86] suggest using a magnitude of relative error (MRE) measure of schedule deviation, or y_i =|(Actual-Planned)/Actual|. Stensrud and colleagues [Stensrud et al. 02] prefer a measure of the magnitude of error relative (MER), or y_i =|(Actual-Planned)/Planned|.

In our nonrandom design, the PAIS dataset itself is a population of assessment cases, where the population is called a *local population* or *a set of available cases* [Lunneborg 00]. Although the PAIS database retains the largest number of assessment cases available anywhere, the dataset is not a random sample, and our results cannot be generalized to all SW-CMM assessments conducted around the world. Hence, interpretation of our results should rightly be limited to assessments reported to PAIS by the current base of CMM users.

Still, it is sensible to make inferences about the *descriptions* to the local population. The descriptions are not inferences to a wider population; rather, they are descriptive statistics which can neither be generalized to others nor have causal implications. Typical descriptions include measures of central tendency (e.g., means or medians), dispersion (e.g., variance or control limits), or relationship (e.g., correlation coefficients or internal consistency).

Descriptions based on a nonrandom sample need assurance that they truly characterize the available cases and that they are stable [Lunneborg 00, Montgomery et al. 98]. An available set of cases such as our assessment dataset cannot be assumed to have the same degree of homogeneity as a random sample. A fair description is a stable one that is relatively uninfluenced by the presence of specific cases. Thus, results such as those in this report should be tested for their stability (homogeneity).

4 Data Analysis

4.1 Correlation and Regression Models

Correlational studies have been used to investigate whether an association exists between increased capability maturity and performance, and under what conditions [Goldenson et al. 99]. All of the previous studies (except case studies) reviewed in Section 2 are correlational studies. In correlational studies, process maturity or capability and performance data from a large number of organizations or projects are collected and statistically analyzed to find relationships between them. Correlational studies typically compute Pearson or Spearman correlations or investigate regression coefficients.

Schedule deviation, as we have defined it, is limited to nonnegative integer values, which can be called count outcomes. More than one regression model exists for count outcomes [Long 97, King 88], so it is necessary to select an appropriate one. The selection should consider the strengths and weaknesses of each model in a specific application field, as well as perceptions in the research community about what are appropriate models of count outcomes.

Schedule deviation is a relatively rare occurrence in our dataset. Many projects reported being less than one month behind schedule, and there are many zero values. Hence, this study uses a zero inflated Poisson (ZIP) regression model.

4.2 A Zero Inflated Poisson (ZIP) Regression Model

ZIP regression has been used elsewhere for predicting count outcomes in software engineering [Khoshgoftaar et al. 02]. The ZIP regression model accounts for the characteristics of an excess number of zero values on the dependent variables, which meets our current needs with schedule deviation. Commonly used Pearson or Spearman correlations are not sufficient to examine such an association.

Our ZIP regression model assumes that the software maintenance processes in an assessed organization are in either a “perfect” or an “imperfect” state. In the perfect state, no schedule deviation will occur, whereas in the imperfect state, there may or may not be schedule deviation. Several factors affect the distribution of schedule deviation in software maintenance and the probability of there being an imperfect state. Process maturity is assumed to be a single factor for the purposes of this study.

Let ψ_i be the probability that the i^{th} maintenance project is performed by a maintenance process that is in a perfect state. Then, $(1-\psi_i)$ becomes the probability that a process of the i^{th} maintenance project is in an imperfect state. Maintenance projects whose processes are in a perfect state are always assumed to be on schedule. Projects whose processes are in an imperfect state may be on schedule following a Poisson distribution with the parameter μ_i, i.e., $\exp(-\mu_i)$. For maintenance processes that are in an imperfect state, the probability that schedule deviation is greater than one month is a product of the probability of being in an imperfect state and the probability of schedule deviation y_i in a Poisson distribution of y_i. Therefore, the probability density function of the ZIP regression model is as follows [Lambert 92, Long 97]:

$$\Pr(y_i \mid x_i) = \begin{cases} \psi_i + (1-\psi_i)\exp(-\mu_i) & \text{for } y_i = 0, \\ (1-\psi_i)\dfrac{\exp(-\mu_i)\mu_i^{y_i}}{y_i!} & \text{for } y_i = 1, 2, \ldots \end{cases} \quad (1)$$

The conditional mean and variance of the ZIP probability function (1) are $\mu_i(1-\psi_i)$ and $\mu_i(1-\psi_i)(1+\mu_i\psi_i)$, respectively. If ψ is 0, then the ZIP regression model (1) becomes a Poisson regression model. The term "conditional" is used to denote that the mean and variance depend on covariates. The only covariate in this study is maturity level.

The ZIP regression model is obtained by the following two link functions:

$$\log(\mu_i) = \beta_0 + \beta_1 \times \text{MATURITY\_LEVEL}$$

$$\text{logit}(\psi_i) = \log\left(\frac{\psi_i}{1-\psi_i}\right) = \gamma_0 + \gamma_1 \times \text{MATURITY\_LEVEL}$$

A negative value of β_1 implies that a high-maturity maintenance process has less schedule deviation than that of a low one. The probability that the maintenance process of project i is in a perfect state is estimated by:

$$\hat{\psi}_i = \frac{\exp(\hat{\gamma}_0 + \hat{\gamma}_1 \times \text{MATURITY\_LEVEL})}{1 + \exp(\hat{\gamma}_0 + \hat{\gamma}_1 \times \text{MATURITY\_LEVEL})}$$

and the Poisson parameter is estimated by

$$\hat{\mu}_i = \exp(\hat{\beta}_0 + \hat{\beta}_1 \times \text{MATURITY\_LEVEL})$$

4.3 Stability Examination

Since our dataset is not a simple random sample, we also need to examine the stability of the analysis results. For this purpose, we use a bootstrap[8] resampling technique that samples B times from the original observation with replacement, where B is a large number such as 1,000. For each sample, the ZIP regression gives interesting descriptions β_1 (coefficients of maturity level in the ZIP regression model), $\mu_i(1-\psi_i)$ (mean), and $\mu_i(1-\psi_i)(1+\mu_i\psi_i)$ (variance). Then, the lower and upper limits of the confidence interval of each description are determined at the 2.5 and 97.5 percentiles respectively from the empirical reference distribution (i.e., a histogram of B replications). The confidence interval of the empirical reference distribution is called the empirical confidence interval (ECI). The bootstrap method is free from unrealistic assumptions such as normality and homogeneity and is suitable to conduct local inferences.

As noted earlier, we use the region of assessed organizations as a mediating contextual factor. The proportions of assessments in the two regions are not fixed in advance; rather, a bootstrap sample is drawn with permutation from the original dataset and then is divided into the U.S. cases or non-U.S. cases before computing our descriptions. Each bootstrap sample is likely to have different proportions of U.S. and non-U.S. cases. This is called "not by design" from the original dataset [Lunneborg 00].

The description from the original dataset should be solidly in the middle of the empirical reference distribution to be considered stable. It should not be at or near the limits of the description. A measure for evaluating stability bias is defined as follows:

$$\text{Bias} = \frac{\sum_{b=1}^{B} t_b^*}{B} - \hat{\theta},$$

where t_b^* is a value of the description at the b^{th} subsample, where $b = 1, \ldots, B$; and $\hat{\theta}$ is a description value from an original dataset.

The degree of bias is evaluated against the standard error (SE) of the description distribution of B replicates. The SE is computed as follows:

$$SE = \sqrt{\frac{\sum_{b=1}^{B}\left(t_b^* - \bar{t}^*\right)}{B-1}}, \text{ where } \bar{t}^* = \sum_{b=1}^{B} t_b^* \Big/ B$$

If the bias is large relative to the SE, there is an instability problem. A criterion for judgment is that if the absolute value of the bias is less than one-quarter of the size of the SE, the bias

[8] This bootstrap method should not be confused with the Bootstrap model for process assessment [Kuvaja 99].

can be ignored [Efron & Tibshirani 93]. Hence, a description from the original dataset can be considered to be stable.

Bootstrap methods have been used previously in empirical software engineering. El-Emam and Garro estimated the number of ISO/IEC 15504 assessments by utilizing a capture-recapture method [El-Emam & Garro 00]. Jung and Hunter utilized a bootstrap method in computing confidence levels for the capability levels for each ISO/IEC 15504 process [Jung and Hunter 01]. Jung and Goldenson used a bootstrap resampling method to evaluate the stability of internal consistency in the SW-CMM [Jung & Goldenson 02].

5 Results

5.1 Descriptive Statistics

This study is based on 752 maintenance projects from 441 SW-CMM CBA IPI assessments. Figure 4 shows the distribution of organizations and maintenance projects by region. A single maintenance project was reported in each of 56 percent of the assessed organizations (171+76=247). Approximately 26 percent of the assessed organizations in the United States included one maintenance project, while about 34 percent of the non-U.S. organizations included a single maintenance project. Two organizations assessed six maintenance projects each. The mean, median, and standard deviation of the number of maintenance projects in these assessed organizations in the United States are 1.67, 1, and 1.01, respectively. In the non-U.S. organizations, the mean, median, and standard deviation of maintenance projects are 1.75, 1.5, and 0.95, respectively.

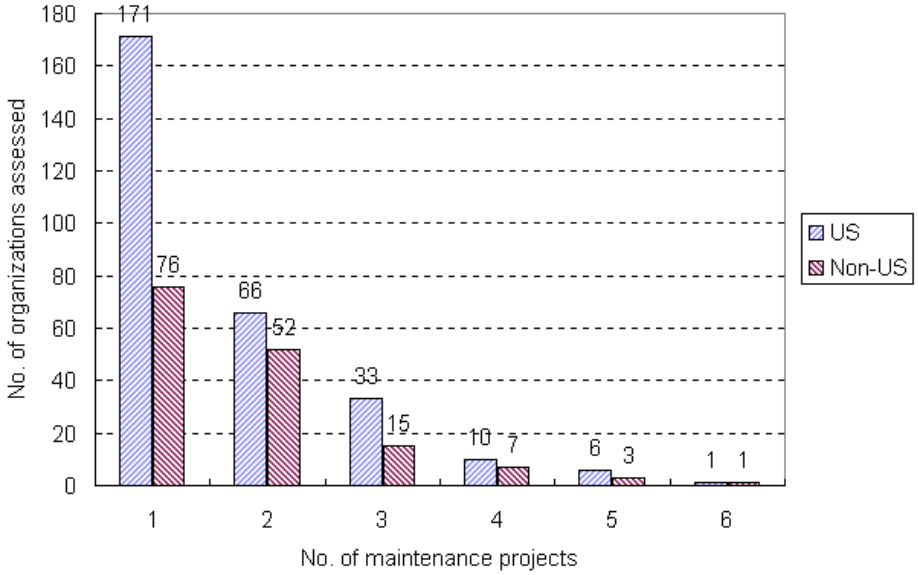

Figure 4: Number of Maintenance Projects in Each Assessed Organization

Figure 5 shows the distribution of maturity level by region. If two or more maintenance projects exist in an assessed organization, the maturity level is counted two or more times. The most frequent maturity level is 2 (Repeatable) in both regions, followed by level 3 (Defined), and level 1 respectively. Means and standard deviations are presented in Table 4. Maturity levels 4 and 5 are not considered in this study because of the very small number of maintenance projects that report delayed schedules at those levels of process maturity.

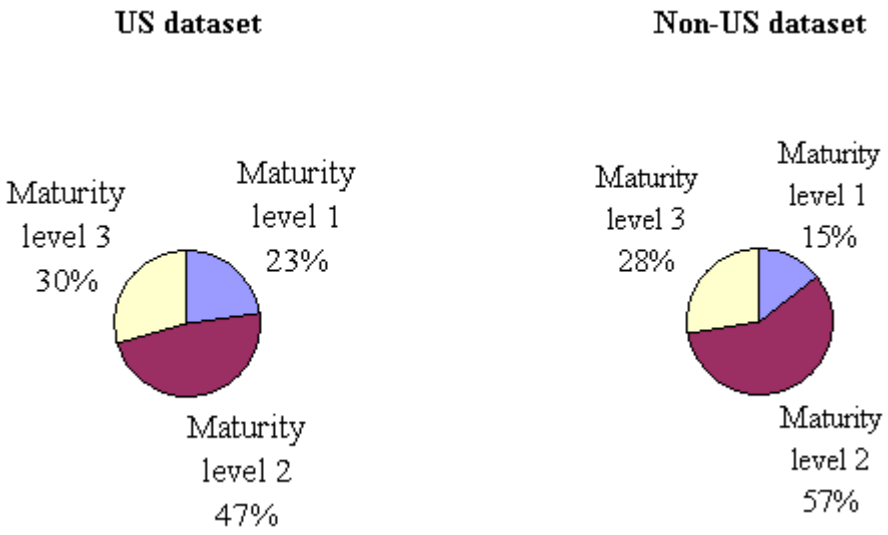

Figure 5: Distribution of Maturity Level Among Assessed Organizations

The proportion of organizations at maturity level 2 clearly is not larger than that at maturity level 1 in software industries throughout the world [Fayad & Laitnen 97]. More likely, as early adopters of a new technology and specifically as organizations interested in software process improvement, the organizations in our sample are drawn from the "higher end" of the maturity spectrum. This phenomenon has been detected in the ISO/IEC PDTR 15504 as well [Rout et al. 98].

As shown in Table 3, the arithmetic mean maturity level in the U.S. dataset is nearly equal to that in the non-U.S. dataset. But, the arithmetic mean of schedule deviations in the U.S. dataset, 0.17, is less than half the value of 0.38 in the non-U.S. dataset.

Table 3: Descriptive Statistics of Maturity Level and Schedule Deviation

	Maturity level		Schedule deviation	
	Mean	Std dev	Mean	Std dev
U.S. (478)	2.07	0.73	0.17	0.99
Non-U.S. (274)	2.13	0.65	0.38	1.44

Table 4 shows the arithmetic mean value of schedule deviance at each maturity level. Though arithmetic means are subject to a lack of robustness, the performance of schedule deviation is improved as maturity level increases in both the U.S. and non-U.S. datasets.

Table 4: Arithmetic Mean of Schedule Deviation at Each Maturity Level

	Maturity level 1		Maturity level 2		Maturity level 3	
	Mean	Std dev	Mean	Std dev	Mean	Std dev
U.S.	0.464	1.750	0.086	0.622	0.069	0.468
Non-U.S.	0.643	2.070	0.407	1.463	0.195	0.828

5.2 Analysis Results

5.2.1 Parameter Estimation and Stability Test

The results of our ZIP regression analyses are given in Table 5. As expected, the estimated coefficient of maturity level, $\hat{\beta}_1$, is both negative and statistically significant for both the U.S. and non-U.S. datasets. The negative association indicates that schedule deviance decreases across the maintenance projects as their respective organizations' maturity levels progressively increase. This is consistent with the hypothesis in our theoretical model.

Table 5: ZIP Regression Results of Schedule Deviation

	U.S.		Non-U.S.	
	Estimated	One-sided p-value	Estimated	One-sided p-value
Intercept (γ_0)	1. 625	0. 002	1. 376	0. 007
MATURITY_LEVEL (γ_1)	0. 682	0. 008	0. 278	0. 089
Intercept (β_0)	1. 886	0	1. 841	0
MATURITY_LEVEL (β_1)	-0. 428	0. 004	-0. 364	0. 009
Goodness-of-fit	χ^2 < 1	1	χ^2 =9. 558	0. 047

In addition, the log ratio of perfect to imperfect state, $\log[\psi_i /(1-\psi_i)]$, has a positive association ($\gamma_1 > 0$) with maturity level. The ratio for the non-U.S. dataset is significant at 8.9%, which indicates only a weak association; however, the results for both regions indicate that the probability of being in a perfect state is increased as maturity progressively increases.

The Chi-square goodness-of-fit values[9] in the last row in Table 5 show the aptness of our ZIP regression model [Cameron & Trivedi 98]. Each of the two fitted models conforms to the assumptions of the ZIP regression model at an alpha value of 1 percent.

Figure 6 shows a graph comparing the fitted and actual probabilities for the non-U.S. case. The better the fit is, the smaller the difference of probabilities. Figure 6 shows that the number of one-month deviation projects is slightly underestimated. On the other hand, the number of projects with three- and four-month deviations is slightly overestimated. However, all of the differences are negligible. For the U.S. dataset, the plot is omitted because the difference of fitted and actual probabilities is quite small.

[9] A null hypothesis for Chi-square goodness-of-fit is that no difference exists between actual counts and estimated counts, i. e. , $\chi^2 = \sum_i (O_i - E_i)^2 / E_i$, where O_i and E_i are observed and estimated schedule deviation, respectively ($E_i \geq 5$). Thus, a large statistic and small p-value implies a poor model fit. The p-value is a right-tail probability [Cameron & Trivedi 98].

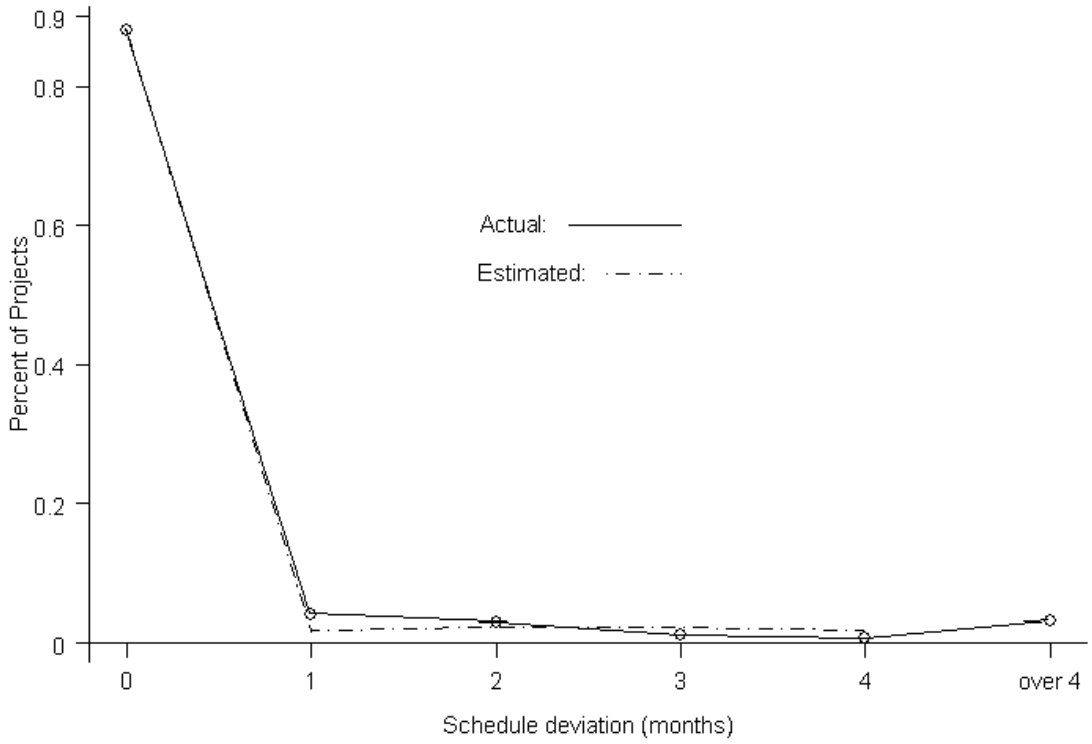

Figure 6: A Plot of Actual and Estimated Probabilities

The estimated coefficients $\hat{\beta}_1$'s are most directly related to our hypotheses, and they are examined for their stability here. Figure 7 shows a bootstrap distribution of the estimated coefficients $\hat{\beta}_1$'s of schedule deviation with 1,000 replicates. For the U.S. dataset (on the left in Figure 7), the dotted and solid vertical lines denote a bootstrap coefficient of –0.463 and an observed coefficient[10] of –0.428 respectively. The difference between them, -0.035, is defined as a bias in bootstrap sampling. It is ignorable in comparison with the SE value of 0.299. Therefore, we conclude that the estimated coefficient $\hat{\beta}_1$ of maturity level is stable. In the bootstrap distribution, 97 percent of the estimated coefficients have negative values.

For the non-U.S. dataset, the bias of the maturity level coefficient, -0.363 - (-0.364) = 0.001, is also ignorable in comparison with the SE value of 0.314. However, 89 percent of the estimates of the maturity level coefficient $\hat{\beta}_1$ are negative. This is a relatively high value compared to the p-value of 0.009 in Table 5.

[10] The term *observed* implies "the sample in our dataset," i.e., the estimated value from our original dataset.

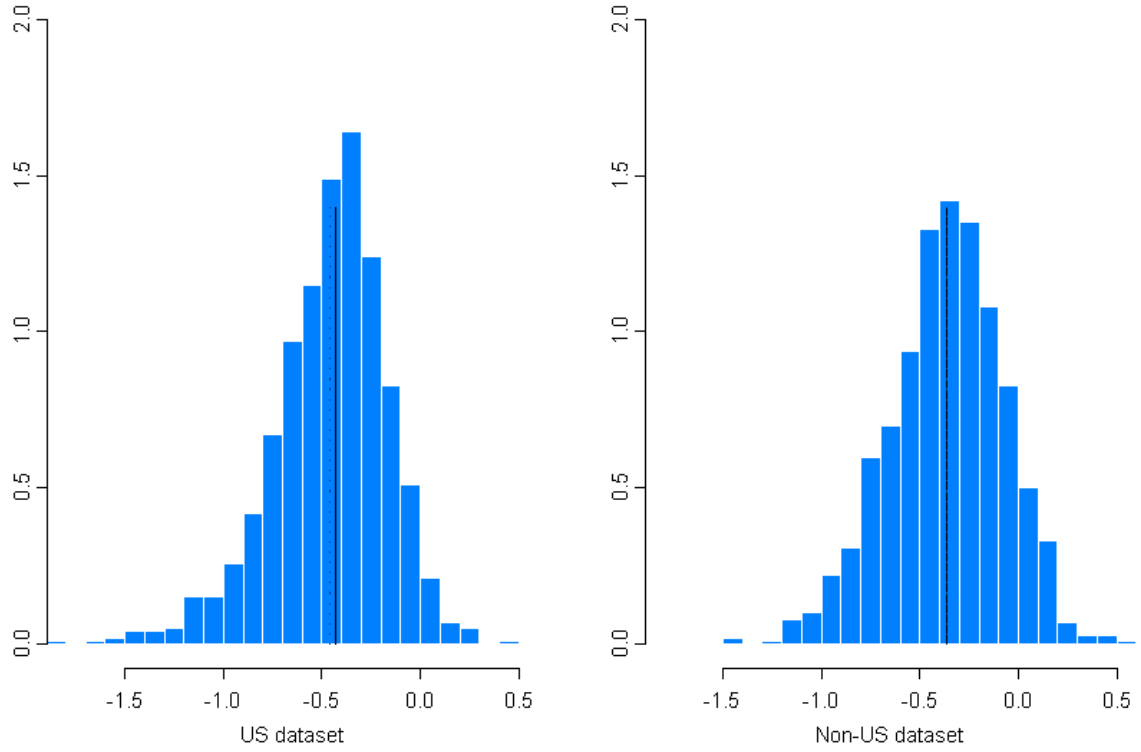

Figure 7: Bootstrap Distribution of Estimated Coefficients $\hat{\beta}_1$

5.2.2 Mean and Its Stability

As seen in Table 5, the negative coefficients $\hat{\beta}_1$ of process maturity support the hypothesis that increases in maturity level result in decreases in schedule deviation. Figure 8 shows the evaluation of HYPOTHESIS 1 in fuller detail. The (expected) mean $\mu_i(1-\psi_i)$ of probability density function (1) is decreasing, and the decrease is distinct for both the U.S. and non-U.S. datasets.

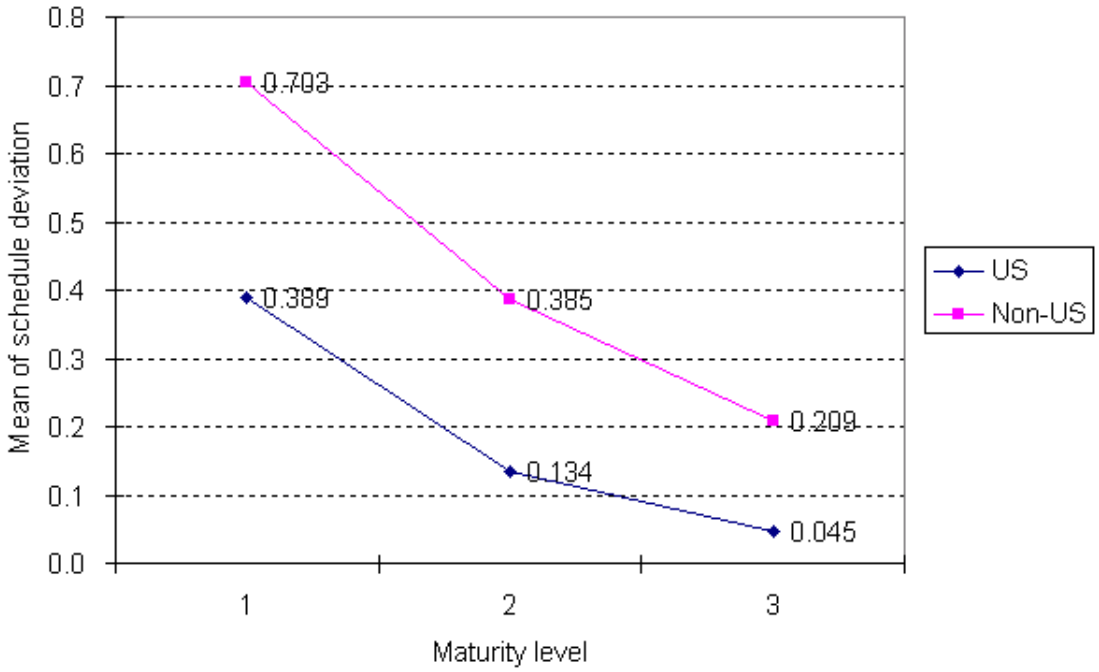

Figure 8: Mean of Schedule Deviation at Maturity Levels 1-3

Table 6 shows results of the bootstrap resampling that examines the stability of the expected mean at each capability level. We conclude that mean at each level is stable because the bias is smaller than one quarter of the SE.

Table 6: Bootstrap Results of Mean Schedule Deviation

Region	Maturity Level	Observed Mean	Bootstrap Mean	Bias	SE	95% ECI
U.S.	1	0.389	0.390	0.001	0.144	[0.134, 0.689]
	2	0.134	0.127	-0.007	0.037	[0.060, 0.206]
	3	0.045	0.047	0.002	0.026	[0.009, 0.108]
Non-U.S.	1	0.703	0.707	0.004	0.265	[0.270.1.273]
	2	0.385	0.374	-0.011	0.086	[0.220, 0.548]
	3	0.209	0.216	0.007	0.084	[0.078, 0.389]

The observed means are a result of the sample in our dataset. Different samples would produce different mean values. Hence, a confidence interval is employed to delimit the true (unknown) mean value of schedule deviation at each maturity level. The 95% ECI in Table 6 is computed from Figure 9, which is a bootstrap empirical reference distribution of mean schedule deviation with 1,000 replicates.

As an example of the ECI interpretation, we can say with a confidence of 95 percent that mean schedule deviation at maturity level 1 in the U.S. dataset is somewhere in the interval

between 0.134 and 0.689. But this interpretation is limited to the current dataset; it cannot be extended to all industries in the United States. Since the bootstrap empirical reference distribution in Figure 9 does not satisfy a normality assumption, using a bootstrap ECI is justified.

Note that there are long tails on the right-hand side of the non-U.S. distributions. They are truncated for reasons of space. In Figure 9, however, the same basic results hold for both the U.S. and non-U.S. data.

The 95% ECIs among the maturity levels in Table 6 partially overlap each other. The empirical reference distributions in Figure 9 also show that overlap. Hence, we must test whether there is a significant difference in the mean schedule deviation between maturity levels. The empirical reference distributions in Figure 9 clearly indicate that we cannot employ a parametric test to examine the mean differences; however, the bootstrap method shows that there are statistically significant difference of mean schedule deviation between maturity levels 1 and 2 and levels 2 and 3 with a p-vale of 0.005 for the both cases; corresponding p-values of 0.04 and 0.039 show that there also are significant differences in mean schedule deviation for the same two pairs of maturity levels in the non-U.S. dataset.

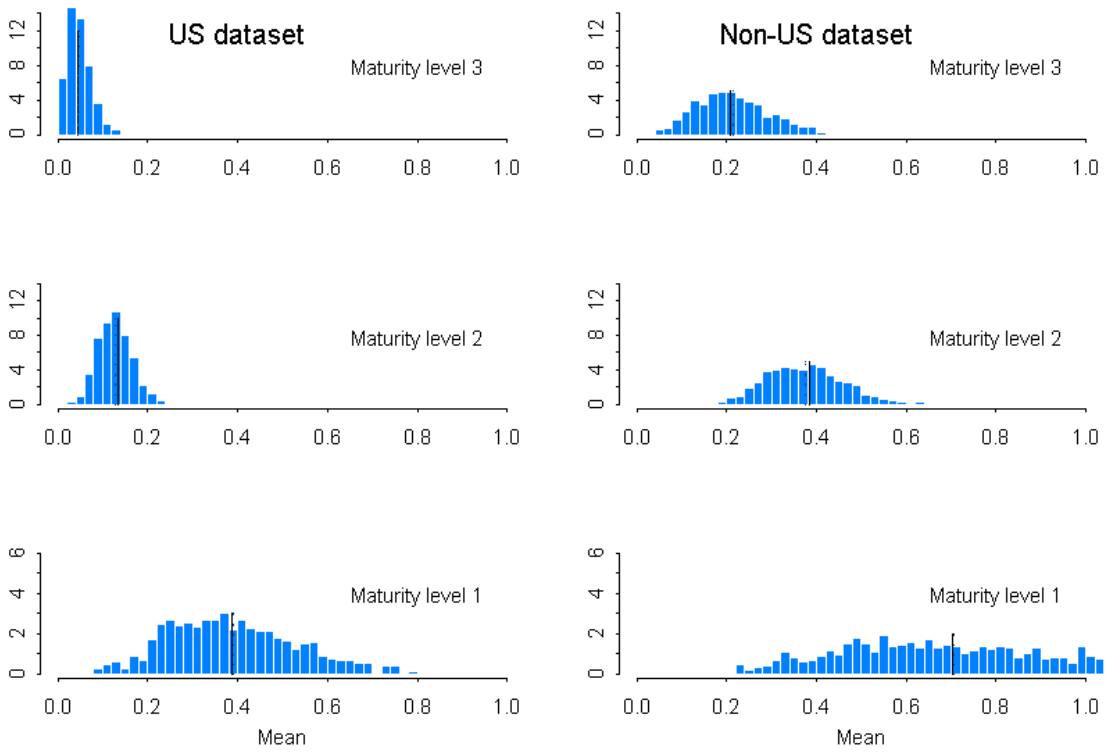

Figure 9: Bootstrap Distribution for Mean Schedule Deviation

5.2.3 Variance and Its Stability

Our second hypothesis requires us to evaluate the reduction of variance in schedule deviation with respect to maturity level. Figure 10 shows how the conditional variance of the ZIP probability density function (1), $\mu_i(1-\psi_i)(1+\mu_i\psi_i)$, is reduced with respect to maturity level. Again, the reduction in variance is significant.

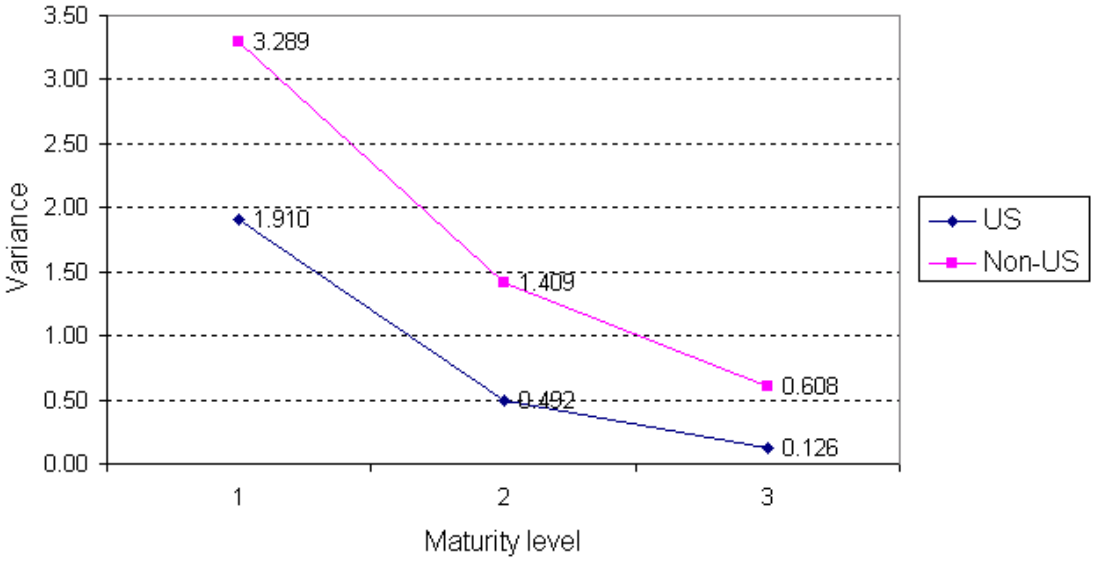

Figure 10: Variance of Schedule Deviation at Maturity Levels 1-3

The results of our bootstrap resampling shown in Table 7 show that the bias is less than a quarter of the SE. The estimated value of variance in schedule deviation also is stable at each maturity level.

Table 7: Bootstrap Results of Variance of Schedule Deviation

Region	Maturity Level	Observed Mean	Bootstrap Mean	Bias	SE	95% ECI
U.S.	1	1.910	1.961	0.051	0.947	[0.523, 4.100]
	2	0.492	0.464	-0.028	0.188	[0.159, 0.905]
	3	0.126	0.138	0.012	0.099	[0.018. 0.370]
Non-U.S.	1	3.289	3.490	0.201	1.909	[0.839, 7.958]
	2	1.409	1.370	-0.039	0.447	[0.647, 2.315]
	3	0.608	0.684	0.076	0.575	[0.133, 1.511]

Finally, the 95% ECIs of conditional variance in Table 7 also are partially overlapped. The empirical reference distributions in Figure 11 lead to the same conclusion. Therefore, we can use the bootstrap empirical reference distributions in Figure 11 to evaluate the variance difference in the schedule deviation between maturity levels. In the U.S. dataset, 95 percent of the 1,000 replicates show that the variance in schedule deviance at maturity level 2 is less

than that at level 1. At maturity levels 2 and 3, the value also is 95 percent. The corresponding values for the non-U.S. dataset are 95 percent and 95 percent respectively. Increases in process maturity are in fact regularly accompanied by reduced variation in schedule deviation by software maintenance projects.

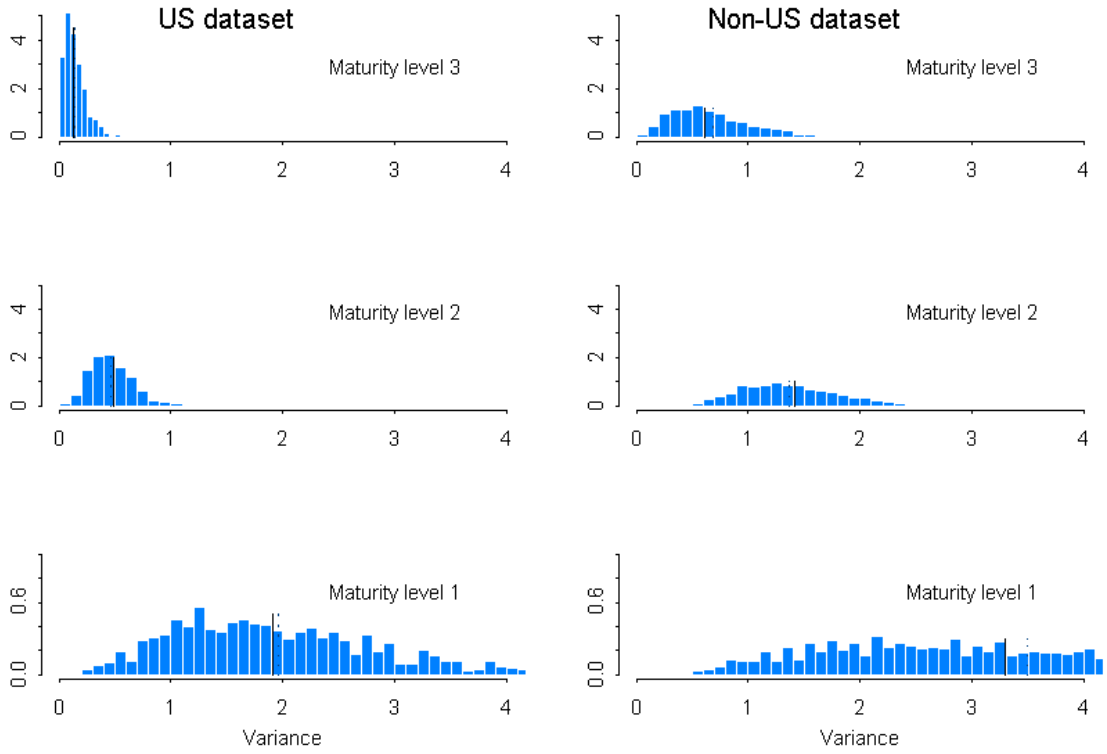

Figure 11: Bootstrap Distribution of Schedule Deviation Variance

6 Conclusion

This study presents compelling evidence about the predictive validity of the SW-CMM as applied to software maintenance. A basic premise of the SW-CMM is that higher maturity should result in better project performance. We find that assessed maturity level is in fact related as expected to schedule deviation in software maintenance projects, and our results are quite robust, in spite of the limitations of the data. While important distinctions remain to be addressed, the results are similar across the software development life cycle; they do not appear to be limited to maintenance projects.

A univariate ZIP regression model is employed to test the premise. Since the results are based on non-random sampling, they are validated using a bootstrap estimation method.

The results show that maintenance projects in higher maturity organizations typically have lower mean and variance in schedule deviation than do comparable projects from organizations assessed at lower levels of maturity. The schedule estimates of projects from higher maturity organizations are markedly more predictably accurate.

Clearly, organizational maturity is not the only factor that affects schedule deviation in software maintenance projects. Neither is schedule deviation the only performance measure worth considering. Other measures of performance such as cost, productivity, quality, and customer satisfaction should be evaluated in future analyses of the predictive validity of Capability Maturity Modeling®. Moreover, such analyses should be extended to CMM Integration℠ and the full life cycle of the development, maintenance, and acquisition of software-intensive systems.

® Capability Maturity Modeling is registered in the U.S. Patent and Trademark Office by Carnegie Mellon University.

℠ CMM Integration is a service mark of Carnegie Mellon University.

References

All URLs are valid as of July 2003.

[Abelson et al. 92] Abelson, R.; Loftus, E.; & Greenwald, A. Ch. 7, "Attempts to Improve the Accuracy of Self-Reports of Voting," 138-153. *Questions About Questions*: *Inquiries into the Cognitive Bases of Surveys,* New York: Russell Sage Foundation, September 1992.

[Alkhatib 92] Alkhatib, G. "The Maintenance Problem of Application Software: An Empirical Analysis." *Journal of Software Maintenance: Research and Practice 1*, 2 (June 1992): 83-104.

[Briand et al. 96] Briand, L.; El-Emam, K.; & Moraska, S. "On the Application of Measurement Theory in Software Engineering." *Empirical Software Engineering: An International Journal 1*, 1 (August 1996): 61-88.

[Briand et al. 00] Briand, L.; Wuest, J.; Daly, J.; & Porter, V. "Exploring the Relationships between Design Measures and Software Quality in Object Oriented Systems." *Journal of Systems and Software 51*, 3 (May 2001): 245-273.

[Brodman & Johnson 02] Brodman, J.; & Johnson, D. "Return on Investment from Software Process Improvement as Measured by U. S. Industry." *CrossTalk 9*, 4 (April 1996): 23-29.

[Butler 95] Butler, K. "The Economic Benefits of Software Process Improvement." *CrossTalk 8*, 7 (July 1995): 14-17.

[Cameron & Trivedi 98] Cameron, A. C. & Trivedi, P. K. *Regression Analysis of Count Data*. Cambridge, UK: Cambridge University Press, 1998.

[Conte et al. 86] Conte, S. D.; Dunsmore, H. E.; & Shen, V. Y. *Software Engineering Metrics and Models*. Menlo Park, CA: Benjamin/Cummings Publishing Company, 1986.

[Diaz & Sligo 97] Diaz, M. & Sligo, J. “How Software Process Improvement Helped Motorola.” *IEEE Software 18*, 3 (Sept/Oct. 1997): 75-81.

[Diaz & King 02] Daiz, M. & King, J. “How CMM Impacts Quality, Productivity, Rework, and the Bottom Line.” *Crosstalk 15*, 3 (March 2002): 9-14.

[Deephouse et al. 95] Deephouse, C.; Goldenson, D.; Kellner, M.; & Mukhopadhyay, T. “The Effects of Software Processes on Meeting Targets and Quality,” 710-719. *Proceedings of the Hawaiian International Conference on Systems Sciences*. Wailea, HI, Jan. 3-6, 1995. Los Alamitos, CA: IEEE Computer Society Press, 1995.

[Deephouse et al. 96] Deephouse, C.; Mukhopadhyay, T.; Goldenson, D.; & Kellner, M. “Software Processes and Project Performance.” *Journal of Management Information Systems 12*, 3 (Winter 1995-96), 187-205.

[Dion 92] Dion, R. “Elements of a Process Improvement Program.” *IEEE Software 9*, 4 (July 1992): 83-85.

[Dion 93] Dion, R. “Process Improvement and the Corporate Balance Sheet.” *IEEE Software 10*, 4 (July 1993): 28-35.

[Drew 92] Drew, D. W. “Tailing the Software Engineering Institute’s (SEI) Capability Maturity Model (CMM) to a Software Sustaining Engineering Organization,” 137-144. *Proceedings of Conference on Software Maintenance*. Orlando, FL, Nov. 9-12. Los Alamitos, CA: IEEE Computer Society Press, 1992.

[Edelstein 93] Edelstein, D. “Report on the IEEE 1219-1993- Standard for Software Maintenance.” *ACM SIGSOFT Software Engineering Notes 18*, 4 (October 1993): 94-95.

[Efron & Tibshirani 93] Efron, B. & Tibshirani, R. J. *An Introduction to the Bootstrap*. New York: Chapman and Hall, 1993.

[El-Emam & Goldenson 95] El-Emam, K. & Goldenson, D. “SPICE: An Empiricist’s Perspective,” 84-97. *Proceedings of the Second IEEE International Software Engineering Standards Symposium*. Montreal, Quebec, Canada, August 21-25, 1995. Los Alamitos, CA: IEEE Computer Society Press, 1995.

[El-Emam & Garro 00] El-Emam, K. & Garro, I. “Estimating the Extent of Standards Use: The Case of ISO/IEC 15504.” *Journal of Systems and Software 53*, 2 (August 2000): 137-143.

[El-Emam & Birk 00a] El-Emam, K. & Birk, A. “Validating the ISO/IEC 15504 Measure of Software Development Process Capability.” *Journal of Systems and Software 51*, 2 (April 2000): 119-149.

[El-Emam & Birk 00b] El-Emam, K. & Birk, A. “Validating the ISO/IEC 15504 Measure of Soft-ware Requirement Analysis Process Capability.” *IEEE Transactions on Software Engineering 26*, 6 (June 2000): 541-566.

[El-Emam et al. 01] El-Emam, K.; Melo, W.; & Machado, J. C. “The Prediction of Faulty Class Using Object-Oriented Design Metrics.” *Journal of Systems and Software 56*, 1 (2001): 63-75.

[Fayad & Laitinen 97] Fayad, M. & Laitinen, M. “Process Assessment Considered Wasteful,” *Communications of the ACM 40*, 11 (November 1997): 125-128.

[Gardner 75] Gardner, P. “Scales and Statistics.” *Review of Educational Research 45*, 1 (Winter 1975): 43-57.

[Glass & Noiseux 81] Glass, R. L. & Noiseux, R. A. *Software Management Guidebook*. Englewood Cliffs, NJ: Prentice-Hall: 1981.

[Goldenson & Herbsleb 95] Goldenson, D. R. & Herbsleb, J. D. *After the Appraisal: A Systematic Survey of Process Improvement, Its Benefits, and Factors That Influence Success*. (CMU/SEI-95-TR-009, ADA302225). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 1995. <http://www.sei.cmu.edu/publications/documents /95.reports/95.tr.009.html>

[Goldenson et al. 99] Goldenson, D. R.; El-Emam, K.; Herbsleb, J.; & Deephouse, C. Ch. 10, “Empirical Studies of Software Process Assessment Methods,” 177-218. *Elements of Software Process Assessment and Improvement*, Los Alamitos, CA: IEEE Computer Society Press, 1999.

[Hall et al. 01] Hall, T.; Rainer, A.; Baddoo, N.; & Beecham, S. “An Empirical Study of Maintenance Issues within Process Improvement Programmes in the Software Industry,” 422-430. *Proceedings of the IEEE International Conference on Software Maintenance*. Florence, Italy, 7-9 Nov. 2001. Los Alamitos, CA: IEEE Computer Society Press, 2001.

[Herbsleb et al. 94] Herbsleb, J.; Carleton, A.; Rozum, J.; Siegel, J.; & Zubrow, D. *Benefits of CMM-Based Software Process Improvement: Initial Results*. (CMU/SEI-94-TR-013, ADA283848). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 1994. <http://www.sei.cmu.edu/publications/documents/94.reports/94.tr.013.html>

[Herbsleb et al. 97] Herbsleb, J.; Zubrow, D.; Goldenson, D.; Hayes, W.; & Paulk, M. “Software Quality and the Capability Maturity Model.” *Communications of the ACM 40*, 6 (June 1997): 30-40.

[Hwang & Jung 03] Hwang, J. & Jung, H-W. “Validating the Process Capability Measures of Project Management Process in ISO/IEC 15504,” 113-119. *Proceedings of SPICE 2003, The Third International SPICE Conference on Process Assessment and Improvement*, Noordwijk, Netherlands, March 17-21, 2003. Noordwijk, Netherlands: ESA, 2003.

[IEEE 90] IEEE. *IEEE Std 610-1990 (IEEE Standard Glossary of Software Engineering Terminology*). New York: Institute of Electrical and Electronics Engineers, 1990.

[ISO/IEC 15504 96] ISO/IEC PDTR 15504, 1996. Part 1-Part 9, Information Technology—Software Process Assessment. Geneva, Switzerland: International Standards Organizations, 1996.

[Jung & Hunter 01] Jung, H-W. & Hunter, R. “The Relationship between ISO/IEC 15504 Process Capability Levels, ISO 9001 Certification and Organization Size: An Empirical Study.” *Journal of Systems and Software 59*, 1 (October 2001): 23-41.

[Jung et al. 01] Jung, H-W.; Hunter, R.; Goldenson, D.; & El-Emam K. “Findings from Phase 2 of the SPICE Trials.” *Software Process Improvement and Practice 6*, 2 (December 2001): 205-242.

[Jung & Goldenson 02] Jung, H-W. & Goldenson, D. The Internal Consistency of Key Process Areas in the Capability Maturity Model® (CMM®) for Software (SW-CMM). (CMU/SEI-2002-TR-037). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 1994. <http://www.sei.cmu.edu/publications/documents/02.reports/02tr037.html>

[Jung & Hunter 02] Jung, H-W. & Hunter, R. "An Evaluation of the SPICE Rating Scale with Regard to the Internal Consistency of Capability Measurement," 105-115. *Proceedings of the SPICE 2002, The Second International SPICE Conference*, Venice, Italy, March 14-15, 2002. Rome, Italy: Nuovo Studio Tecna, 2002.

[Kajko-Mattsson 02] Kajko-Mattsson, M. "Problem Management Maturity within Corrective Maintenance." *Journal of Software Maintenance: Research and Practice 14*, 3 (May-June 2002): 197-227.

[Kemerer 95] Kemerer, C. F. "Software Complexity and Software Maintenance: A Survey of Empirical Research." *Annals of Software Engineering 1* (September 1995):1-22.

[Khoshgoftaar et al. 02] Khoshgoftaar, T. M.; Geleyn, E.; & Gao, K. "An Empirical Study of the Impact of Count Models Predictions on Module-Order Models," 161-172. *Proceedings of the Eighth IEEE Symposium on Software Metrics*, Ottawa, Canada, June 4-7, 2002. Los Alamitos, CA: IEEE Computer Society Press, 2002.

[King 88] King, G. "Statistical Models for Political Science Event Counts: Bias in Conventional Procedures and Evidence for the Exponential Poisson Regression Model." *American Journal of Political Science 32*, 3 (August 1988): 838-863.

[Krasner 99] Krasner, H. Ch. 9, "The Payoff for Software Process Improvement: What it is and How to Get it." 151-176. *Elements of Software Process Assessment and Improvement*, Los Alamitos, CA: IEEE Computer Society Press, 1999.

[Krishnan & Kellner 99] Krishnan, M. S. & Kellner, M. I. "Measuring Process Consistency: Implications for Reducing Software Defects." *IEEE Transactions on Software Engineering 25*, 6 (November 1999): 800-815.

[Kuvaja 99] Kuvaja, P. "Bootstrap 3.0–A SPICE Conformant Software Process Assessment Methodology." *Software Quality Journal 8,* 1 (September 1999): 7-19.

[Kuilboer & Ashrafi 00] Kuilboer, J. P.; & Ashrafi, N. "Software Process and Product Improvement: An Empirical Assessment." *Information and Software Technology Journal 42,* 1 (January 2000): 27-34.

[Lambert 92] Lambert, D. "Zero-Inflated Poisson Regression, with an Application to Defects in Manufacturing." *Technometrics 34*, 1 (February 1992): 1-14.

[Lawlis et al. 96] Lawlis, P.; Flowe, R.; & Thordahl, J. "A Correlational Study of the CMM and Software Development Performance." *Software Process Newsletter* 7, (Fall 1996): 1-5.

[Lerner 94] Lerner, M. "Software Maintenance Crisis Resolution: The New IEEE Standard." *Software Development 2*, 8 (August 1994): 65-72.

[Lientz & Swanson 80] Lientz, B. & Swanson, E. *Software Maintenance Management*. Reading, MA: Addison-Wesley, 1980.

[Loftus et al. 92] Loftus, E. F.; Smith, K. D.; Klinger, M. R.; & Fielder, J. Ch. 6, "Memory and Mismemory for Health Events," 102-137. *Questions About Questions: Inquiries into the Cognitive Bases of Surveys*, New York: Russell Sage Foundation, September 1992.

[Long 97] Long, J. S. "Regression Models for Categorical and Limited Dependent Variables." Sage University Paper Series on Advanced Quantitative techniques in the Social Sciences, 07, London UK, 1997.

[Lunneborg 00] Lunneborg, C. *Data Analysis by Resampling: Concepts and Applications*. Pacific Grove, CA: Duxbury, Thomson Learning, 2000.

[McCallum et al. 95] McCallum, J.; Raymond, C.; & McGilchrist, C. "How Accurate are Self Reports of Doctor Visits? A Comparison Using Australian Health Insurance Commission Records." National Center for Epidemiology and Population Health. The Australian National University, 1995.

[McGarry et al. 98] McGarry, F.; Burke, S.; & Decker, B. “Measuring the Impacts Individual Process Maturity Attributes Have on Software Projects,’’ 52-60. *Proceedings of the 5th International Software Metrics Symposium*. 1998. Bethesda, MD, Nov. 20-21, 1998. Los Alamitos, CA: IEEE Computer Society, 1998.

[Montgomery et al. 98] Montgomery, D. C.; Runger, G. C.; & Hubele, N. F. *Engineering Statistics*. New York: John Wiley & Sons, Inc., 1998.

[Niessink & van Vliet 00] Niessink, F. & van Vliet, H. “Software Maintenance from a Service Perspective.” *Journal of Software Maintenance: Research and Practices 12*, 2 (March-April 2000): 103-120.

[Nunnally & Bernstein 94] Nunnally, J. C. & Bernstein, I. H. *Psychometric Theory*. New York: McGraw-Hill, Inc., 1994.

[Paulk et al. 93a] Paulk, M.; Curtis, B; Chrissis, M. & Weber, C. *Capability Maturity Model for Software, Version 1. 1* (CMU/SEI-93-TR-024, ADA 263403). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, February 1993. <http://www.sei.cmu.edu /publications/documents/93.reports/93.tr.024.html>

[Paulk et al. 93b] Paulk, M.; Weber, C.; Garcia, S.; Chrissis, M. & Bush, M. *Key Practices of the Capability Maturity Model, Version 1. 1* (CMU/SEI-93-TR-025, ADA263432). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, February 1993. <http://www.sei.cmu.edu/publications/documents/93.reports /93.tr.025.html>

[Paulk et al. 93c] Paulk, M.; Curtis, B.; Chrissis, M.; & Weber, C. “Capability Maturity Model, Version 1.1.” *IEEE Software 10*, 4 (July 1993): 18-27.

[Paulk et al. 96] Paulk, M.; Weber, C.; Curtis, B.; & Chrissis, M. B. *The Capability Maturity Model: Guidelines for Improving the Software Process*. New York: Addison-Wesley, 1996.

[Paulk 99] Paulk, M. “Analyzing the Conceptual Relationship Between ISO/IEC 15504 (Software Process Assessment) and the Capability Maturity Model for Software,” 293-303. *Proceedings, Ninth International Conference on Software Quality*, Cambridge, MA, Oct. 4-6, 1999. Milwaukee, WI: American Society for Quality Control, 1999.

[Perry et al. 96] Perry, D.; Staudenmayer, N.; & Votta, Jr. L. Ch. 5, "Understanding and Improving Time Usage in Software Development," 111-135. *Software Process*, New York: John Wiley & Sons, Inc., 1996.

[Rout et al. 98] Route, T.; Tuffley, A.; & Hodgen, B. "Process Capability in the Australian Software Industry: Results from the SPICE Trials," 12-19. *Australian Software Engineering Conference (ASWEC'98*). Adelaide, Australia, 1998. IEEE Computer Society: Los Alamitos, CA, 1998.

[Schrank et al. 95] Schrank, M.; Anderson, A.; Bisignani, M; & Boyce, G. Jr. "Assessing the Capability of Military Software Maintenance Organizations," 314-319. *Proceedings of the 14th Digital Avionics Systems Conference* (DASC'95), 1995. New York: IEEE Computer Society, 1995.

[SEI 02] SEI. *Process Maturity Profile of the Software Engineering Community*. <http://www.sei.cmu.edu/sema/profile.html>

[Stensrud et al. 02] Stensrud, E.; Foss, T.; Kitchenham, B.; & Myrtveit, I. "An Empirical Validation of the Relationship Between the Magnitude of Relative Error and Project Size," 3-12. *Proceedings of the Eighth IEEE Symposium on Software Metrics*, Ottawa, Canada, June 4-7, 2002. Los Alamitos, CA: IEEE Computer Society Press, 2002.

[Stevens 51] Stevens, S. Ch. 1, "Mathematics, Measurement, and Psychophysics," 1-49. *Handbook of Experimental Psychology*. New York: John Wiley & Sons, Inc., 1951.

[Swanson & Beath 89] Swanson, E. B. & Beath, C. M. *Maintaining Information Systems in Organizations*. New York: Halsted Press, 1989.

[Velleman & Wilkinson 93] Velleman, P. & Wilkinson, L. "Normal, Ordinal, Interval, and Ratio Typologies Are Misleading." *The American Statistician 47*, 1 (February 1993): 65-72.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
(Leave Blank)	July 2003	Final

4. TITLE AND SUBTITLE	5. FUNDING NUMBERS
CMM®-Based Process Improvement and Schedule Deviation in Software Maintenance	F19628-00-C-0003

6. AUTHOR(S)

Ho-Won Jung, Dennis R. Goldenson

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)	8. PERFORMING ORGANIZATION REPORT NUMBER
Software Engineering Institute Carnegie Mellon University Pittsburgh, PA 15213	CMU/SEI-2003-TN-015

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSORING/MONITORING AGENCY REPORT NUMBER
HQ ESC/XPK 5 Eglin Street Hanscom AFB, MA 01731-2116	

11. SUPPLEMENTARY NOTES

12A DISTRIBUTION/AVAILABILITY STATEMENT	12B DISTRIBUTION CODE
Unclassified/Unlimited, DTIC, NTIS	

13. ABSTRACT (MAXIMUM 200 WORDS)

The objective of this study is to evaluate the predictive validity of the Capability Maturity Model® (CMM®) for Software (SW-CMM) as applied to software maintenance.

The SW-CMM is intended to apply to both software development and maintenance. A basic premise (hypothesis) of the SW-CMM is that improving process maturity will result in better project performance and product quality. The extent to which that hypothesis is supported empirically is called a test of its predictive validity. No previous evaluation exists of the predictive validity of the SW-CMM in a maintenance context.

The extent to which schedule estimates differ from reality is one important measure of project performance. But is higher maturity in fact correlated with a reduction in schedule deviation? Data from 752 maintenance projects drawn from 441 SW-CMM assessments are analyzed using a zero inflated Poisson (ZIP) regression model, and the results are validated using a bootstrap estimation method. Projects from higher maturity organizations typically report less schedule deviation than those from organizations assessed at lower maturity levels.

14. SUBJECT TERMS	15. NUMBER OF PAGES
SW-CMM, predictive validity, schedule deviation, software maintenance, project improvement, process improvement	46

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	UL

NSN 7540-01-280-5500 Standard Form 298 (Rev. 2-89) Prescribed by ANSI Std. Z39-18 298-102

® Capability Maturity Model and CMM are registered in the U.S. Patent and Trademark Office by Carnegie Mellon University.