Empirical Studies of Software Process Assessment Methods*†

Dennis R. Goldenson
Software Engineering Institute
Carnegie Mellon University
Pittsburgh, PA 15213-3890
U.S.A.
dg@sei.cmu.edu

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

James Herbsleb
Software Engineering Institute
Carnegie Mellon University
Pittsburgh, PA 15213-3890
U.S.A.
jherbsle@sei.cmu.edu

Christopher Deephouse
NetBill[tm]
Hamburg Hall A222
Carnegie Mellon University
Pittsburgh, PA 15213-3890
U.S.A.
cdeephouse+@cmu.edu

1. Introduction

There are now many methods for assessing the maturity and capabilities of software engineering organizations. Assessment scores are being used in making the contract award decision by the U.S. Navy [Ru93] and Air Force [SK95], as well as in commercial organizations [MC96]. Furthermore, conformance to process standards such as ISO 9001, as determined during an audit, is a necessity for doing business in many European countries. Software process assessments are also an essential element of the self-improvement cycle for many organizations (e.g., see[Ba96][Dy95]).

There has been a relative dearth of empirical investigations of the core premises of most contemporary assessment methods and their underlying models. Software organizations were being required and/or pressured to conform to certain standards (e.g., to be at Level 3 on the CMM) without adequate empirical evidence supporting the assumptions made by these standards. At least partly because of this, a certain amount of skepticisim and uncertainty exists about the accuracy and usefulness of software process assessments, and improvements based on them (e.g., see [BM91][Ba94][Ba95][Jo95]). The software community needs to be more confident that assessment results accurately reflect the capabilities of organizations being assessed, not simply the idiosyncrasies of those doing the assessments. We need a solid basis to better understand assessment methods, evaluate their basic premises, and inform decisions about their use and improvement. Similarly, more evidence is needed to justify investment in process improvement programs following the assessments.

Despite criticisms on lack of evidence, by now, a good number of empirical studies in fact do exist. Hence, the objectives of this chapter are twofold: (a) to demonstrate the different approaches for empirically studying software process assessment methods and the impact of software process on subsequent performance, and (b) to summarize the results of some empirical studies of software process assessment methods that have been conducted to date. Following this introduction, the chapter consists of three sections. Validity issues are addressed in Section 2, which examines the extent to which assessment methods are really measuring best software engineering practices. Reliability issues are

---

* Work at the SEI is sponsored by the U.S. Department of Defense. Work done by El Emam in the SPICE project has been supported, in part, by the Applied Software Engineering Centre (ASEC) in Montreal.

† To appear in *Software Process Assessment and Improvement*, T. P. Rout (ed.), published by Computational Mechanics, 1997. Also appears as International Software Engineering Research Network technical report ISERN-97-09.

addressed in Section 3, which examines the extent to which assessment scores and profiles are repeatable and consistent. Illustrative examples are drawn from recent empirical work conducted by the authors, including studies on the CMM[sm] , and from field trials of the emerging SPICE[1] standard. We conclude in Section 4 with directions for future research.

2. Validity

Validity of measurement is defined as the extent to which a measurement procedure is measuring what it is purporting to measure [Ker86]. During the process of validating a measurement procedure one attempts to collect evidence to support the types of inferences that are to be drawn from measurement scores. In the context of software process assessments, concern with validity is epitomized by the question: "are software process assessments really measuring best software process practices?"

There are a number of different types of validity that have been defined in the behavioral sciences measurement literature: content validity, concurrent validity, predictive validity, and construct validity. These are summarized in Figure 1[2] (also see [EG95]).

The software process community has been demanding objective empirical evidence showing that increased maturity or capability, in fact, leads to greater project and organizational effectiveness (measured in various ways, such as productivity and/or product quality). Indeed, predictive validity can be considered to be the most important type of validity. For example, in a 1993 IEEE Software article [Her93], Hersh states *"despite our own firm belief in process improvement and our intuitive expectation that substantial returns will result from moving up the SEI scale - we still can't prove it."* Furthermore, in surveys[3] of software process improvement professionals conducted by the SEI in 1993 [HC+94], when asked about their most pressing needs, *"The top-ranked need reported by survey respondents was the need for quantitative information regarding the benefits from software process improvement efforts."*

[sm] CMM and capability Maturity Model are service marks of Carnegie Mellon University.

[1] SPICE (Software Process Improvement and Capability dEtermination) is an international project established by the International Organization for Standards (ISO) and the International Electrotechnical Commission (IEC) to deliver an international standard for software process assessment. As part of this project there is an empirical trials phase [EG95][MO95]. More information about SPICE may be found in [Do93][Dr94][Dr95][KBD96][Ko94][PK94][Ro95].

[2] It should be noted that different authors give different names to the definitions we have provided, and others will give different definitions to the types of validities that are presented. However, the definitions we have used are quite common.

[3] These surveys were conducted by Goldenson at the SEI, but the detailed findings were not made public.

Type of Validity	Definition	Method of Validation
Content Validity	Representativeness or sampling adequacy of the content of a measuring instrument	Largely expert judgement
Concurrent Validity	Equivalence of scores obtained from different assessment methods	Correlation (or other measures of association) between the scores obtained from the different methods
Predictive Validity	Utility of software process assessment scores in predicting future project and/or organizational performance	Correlation (or other measures of association) between software process assessment scores and some criterion measure of effectiveness
Construct Validity	The extent to which different software process assessment methods measure the same concept	• MultiTrait-MultiMethod Matrix (see [EG95]) • Factor Analysis (see [Ker86][Nu78])

Figure 1: The different types of validity.

In response to such demands, two classes of empirical studies have been conducted and reported: case studies and correlational studies. Case studies describe the experiences of a single organization (or a small number of selected organizations) and the benefits it gained from increasing its maturity level. Case studies are most useful for showing that there are organizations that have benefited from increased maturity. Examples of these are reported in [HSW91][HC+94][Di92][Di93][WR93][BF95][LB92][Bu95] [Leb96]. However, in this context, case studies have a methodololgical disadvantage that makes it difficult to generalize the results from a single case study or even a small number of case studies. Case studies tend to suffer from a selection bias because:

- 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 (e.g., *all* the references to case studies above are success stories), and

- 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.

With correlational studies, one collects data from a number of organizations and investigates relationships between maturity and organizational and/or project effectiveness statistically. Correlational studies are useful for showing whether a general association exists between increased maturity and effectiveness, and under what conditions. We describe here the results of four recent survey based studies that do provide comparative evidence about the impact of software process improvement. Of course, sample surveys have several limitations, but they do provide needed comparison in a classic trade off of depth for breadth.

Survey data are often faulted for being based only on opinion and it is difficult to frame meaningful questions that convey shared meaning to different people. However, as implied by the backgrounds and experience of the respondents in these surveys, they can be expected to provide reasonably informed opinions. Of course we present here only a handful of studies. We need many more efforts that focus in

more depth on the results of different, more specific aspects of software process improvement, and that do so in differing contexts under different circumstances of process maturity, domain, and complexity. Still, the present results are quite compelling.

		Unit of Analysis		
		Life Cycle Process	**Project**	**Organization**
Type of Criterion Variable	**Life Cycle Process Effectiveness**	"User Participation in the Requirements Engineering Process" (Section 2.4)		"Effect of Maturity on Individual Processes: The Case of Requirements Engineering" (Section 2.2)
	Project Effectiveness		"Does Software Process Improve Project Results?" (Section 2.3)	
	Organizational Effectiveness			"What Happens After the Assessment?" (Section 2.1)

Figure 2: Types of Correlational Studies.

The four surveys reviewed below represent different approaches that can be used to examine the validity of maturity measures. The differences among them are in the *unit of analysis* of the study and in the choice of *the type of the criterion variable(s)*. A classification of the four studies along these two dimensions is given in Figure 2.

- The first approach is the most direct one. In this chapter, we present the results from a survey of individuals in organizations that have been appraised using the CMM[sm] and investigated the relationship between organizational maturity and organizational effectiveness (Section 2.1).

- One can also investigate the relationship between the results from assessing organizational maturity and the effectiveness of individual life cycle processes. Here, one is looking for evidence demonstrating a positive relationship between maturity and the effectiveness of individual life cycle processes.The survey we present in this chapter used the success of the requirements engineering process as an outcome variable (Section 2.2).

- People involved in projects can be asked questions about the effectiveness of individual practices that are also usually evaluated during assessments. If there is evidence supporting the effectiveness of these practices, then they are justifiaby evaluated during assessments. The survey we present here used measures of project effectiveness (Section 2.3).

- One survey investigated under what conditions user participation is effective in the requirements engineering process. If there is evidence supporting the effectiveness of user participation, then it can be justifiaby evaluated during assessments as an indicator of maturity/capability. The survey we present in this chapter used the success of the requirements engineering process as an outcome variable (Section 2.4).

These different approaches provide us with wider perspectives on how increased maturity and capability have an impact on effectiveness.

2.1 What Happens After the Assessment?

2.1.1 The Study

Results from a survey of organizations that have undergone process assessments based on the Capability Maturity Model for Software (CMM[sm]) provide important new evidence both about the value of the assessments themselves and about the payoff that can be expected from software process improvement [GH95][HG96]. In that study, the authors tracked the experiences of a large number of software organizations over an extended period of time since their assessments.

As seen more fully in [GH95][HG96], the survey was based on the responses of 138 individuals from 56 software organizations in the United States and Canada. Completed questionnaires from 83 percent were received from those to whom questionnaires were sent, representing 92 percent of the organizations about which the authors were able to obtain point of contact information.

Their respective Software Process Assessments (SPAs) took place approximately one to three years prior to the survey -- long enough for genuine change to have taken place, yet recent enough to expect accurate recall from people familiar with the assessments and their aftermaths. It was attempted to include as broad a cross section of organizations as possible in the sample. It includes organizations that vary in size and sector within the software industry, and that vary in the success they have experienced in their process improvement efforts.

People who fill different roles in an organization might be expected to differ in their perspectives about the same events. In particular, those who are personally invested in process improvement might be accused of bias in favor of their own efforts. Hence, in addition to an organizational level software engineering process group (SEPG) manager or someone with equivalent responsibilities, a project level software manager and a well-respected senior developer or similar technical person were sought from each organization. Interestingly enough, there were no consistent, statistically significant differences among these individuals' answers to the questions that were posed in the survey.

2.1.2 Results

Progress Since the Assessments: Overall, the respondents believe that their asessments were more than worthwhile. Close to three-quarters of them agreed to the rather strongly worded statement that "the assessment was well worth the money and effort we spent; it had a major positive effect on the organization."

Indeed, a substantial amount of progress appears to have taken place since these assessments were conducted (Figure 3). The vast majority of the respondents report having followed up their assessments with action plans and process action teams to carry out those plans. Over 70 percent said that their organizations had implemented process changes in demonstration projects and/or organization-wide as a result of their assessments.

Not all of the post-assessment activity was equally effective (Figure 4). However, well over half of the respondents stated that their organizations had experienced at least moderate success in addressing the findings and recommendations that were raised by their assessments; almost one-third said there had been substantial success or marked success throughout their organizations. Only 14 percent say they had had little if any appreciable success by the time of the survey.

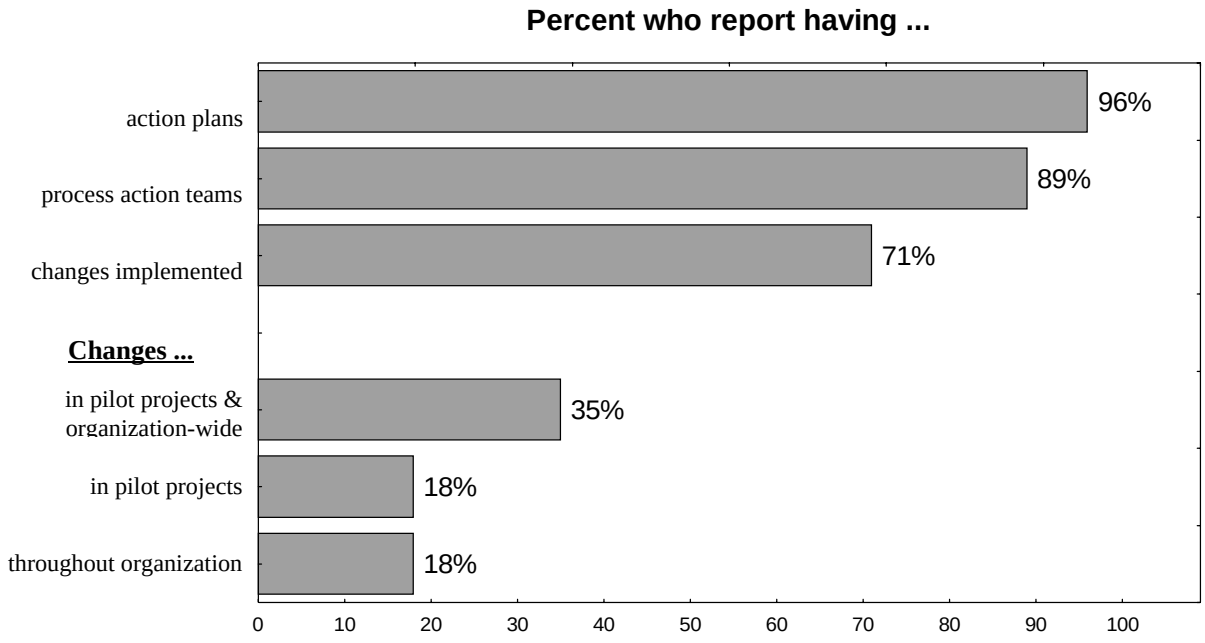


Figure 3: Post-appraisal activities.

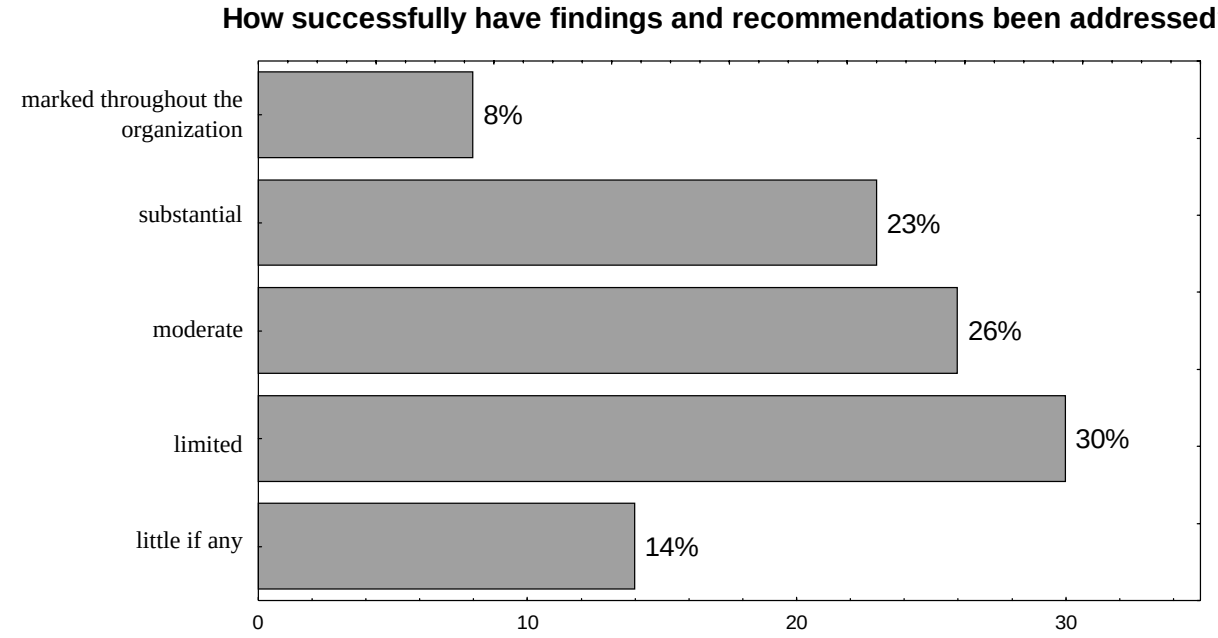


Figure 4: Success in addressing appraisal results.

Contrary to some critics, there is very little evidence that the assessments and subsequent process improvements efforts had a negative effect. Very few (4 percent) respondents said that their assessments had been counter-productive. Over 80 percent said that software processes had not become more bureaucratic and that technical creativity had not been stifled. In fact, there is evidence that more mature organizations in the commercial and government sectors actually have fewer paperwork requirements than do less mature organizations.

Finally, as seen more fully in [GH95][HG96], most of the survey respondents reported that their assessments were both accurate and useful for their process improvement efforts. As just seen, not all organizations were equally successful. Improvement also often took longer and cost more than people had expected. However several factors, most of them under management control, do distinguish among the organizations whose improvement efforts varied in success.

Benefits of Process Improvement: These data suggest that process maturity does typically pay off in better organizational performance than would otherwise be expected. Survey respondents from higher maturity organizations are substantially more likely than those from level 1 organizations to report that their organizations are characterized by high product quality and staff productivity. They claim better ability to meet schedule commitments, and to have higher levels of staff morale and job satisfaction. In addition they tend to report doing better with respect to customer satisfaction and success in meeting budget commitments. The basic results appear to be unaffected by organizational size. Moreover, they hold for organizations from different sectors of the software industry, not just among those from defense contractors and the federal government.

As seen in the graphs in Figure 5, respondents from higher maturity organizations are more likely than those who remain at the Initial Level to characterize performance in their organizations as being "good" or "excellent" as opposed to just "fair" or "poor." Five of the six correlations are statistically significant (at the 0.05 level according to chi-square criteria). The sixth, ability to meet budget commitments, in fact approaches statistical significance.

The overall patterns in Figure 5 are quite clear[4]. For example, 80 percent of those from level 3 organizations said that their ability to meet schedule is "good" or "excellent". Fewer than half as many (39 percent) at the initial level make a comparable claim.

Notice also the pattern of responses about product quality. Not surprisingly, people from all maturity levels tend to report that their organizations do a reasonably good job of providing their customers with high quality products. Even so, almost one-fourth of the level 1 respondents admit that their products are of only "fair" or "poor" quality[5]. Yet all of the respondents whose organizations have achieved level 3 report having products of good or excellent quality. In fact (not shown in the Figure), close to two-thirds at level 3 say that their products are of excellent quality. Fewer than ten percent of the level 1 or level 2 respondents make a similar claim.

Staff morale also appears to benefit quite substantially by higher process maturity. As seen in Figure 5, 60 percent of those whose organizations are at the defined level report that their staff morale is good or excellent. Fewer than a quarter (23 percent) of those at the initial level report that morale is good (only one says it is excellent); another 23 percent (not shown in the Figure) say that their morale level is poor.

Not unlike the situation with respect to product quality, people from all maturity levels tend to report that their organizations have achieved reasonably high levels of customer satisfaction. However there is an unexplained dip in reported customer satisfaction at the repeatable level. Although the overall pattern of responses is statistically significant, the difference between the initial and repeatable organizations is not. Still, one can conjecture about the reasons for the dip in these data. Upon achieving level 2, some organizations very well may have to deal with new expectations about the conditions under which requirements can be changed. Customer satisfaction may suffer, at least temporarily, as a result.

---

[4] The data in Figure 5 are based on the respondents' self-reports of their maturity levels at the time of the survey. It was assumed that at least some organizations would have achieved higher maturity levels within a period of one to three years after their appraisals [HZ95]. However, there was in fact relatively little difference between the actual appraisal and self-reported maturity levels, and correlations based on both measures of maturity are similar.

[5] The percentage exceeds one-third when those who said that their organizations were approaching level 2 by the time of the survey were removed.

However, it may simply be that the respondents from level 1 organizations tend to overstate their success in keeping their customers satisfied.

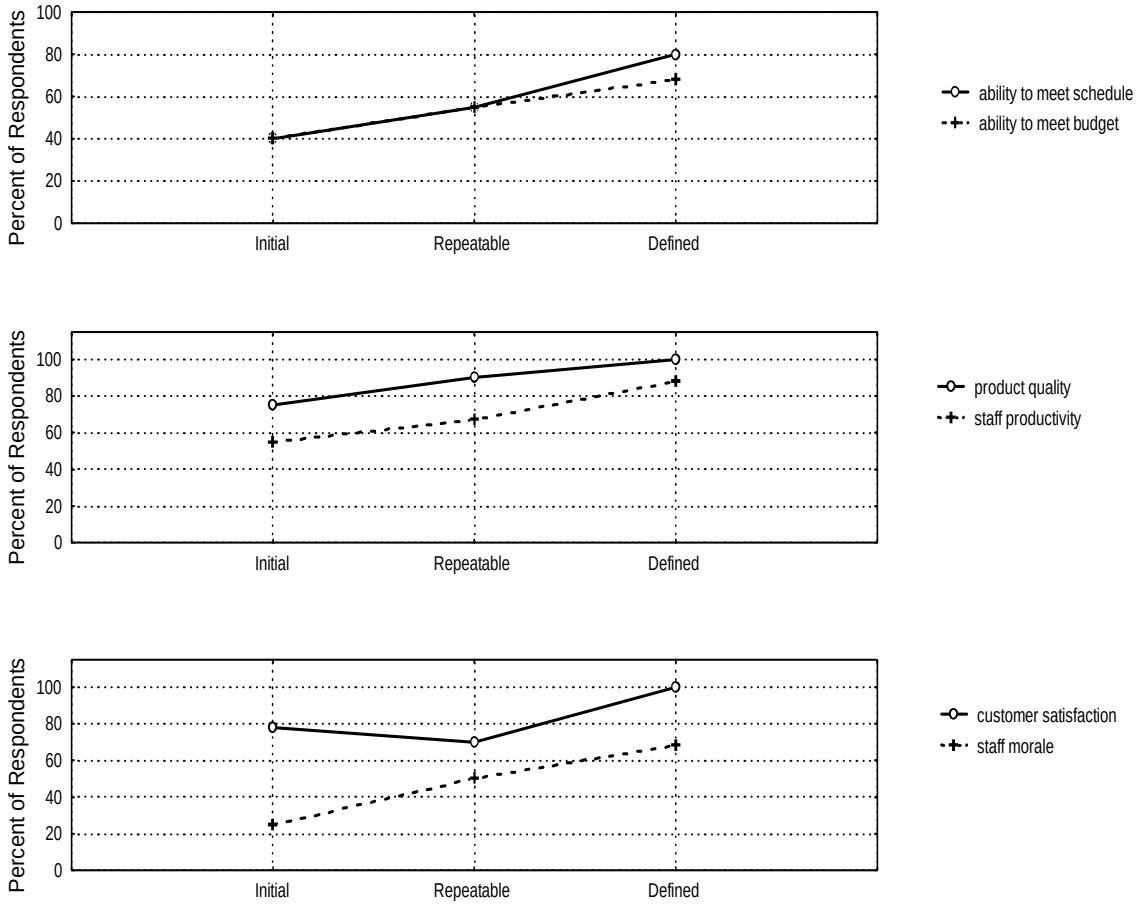


Figure 5: Impact of SPI on organizational performance.

2.2 Effect of Maturity on Individual Processes: The Case of Requirements Engineering

2.2.1 The Study

The study by El Emam and Madhavji [EM95a] investigated the relationship between organizational maturity and the success of the requirements engineering (henceforth RE) process in MIS organizations. The RE process is considered to be one of the more critical life cycle processes. For example, previous empirical studies have shown that it costs much less to fix defects during the RE process than during later phases of the life cycle, that the number of defects in requirements documents is related to software errors, and that the source of the majority of defects found in code is the RE phase. Given the above evidence, it is reasonable to assume that if organizational maturity strengthens the RE process, then this

would also strengthen processes later in the life cycle and their corresponding products. It was therefore hypothesized that there would be a positive association between the maturity of MIS organizations and the success of the RE process for a selected project in the organization.

Four dimensions of organizational maturity were measured: (a) standardization, which is concerned with process and product standardization in the MIS organization, (b) project management, which is concerned with the extent to which good project management practices are employed, (c) tools, which is concerned with effective automated tool usage in the organization, and (d) organization, which is concerned mainly with the documentation of the overall organization's missions and goals and the alignment of the MIS organization with these. The measures of these dimensions were based on the concepts in contemporary maturity models, such as the CMM[sm], as well as an earlier stage model developed by Nolan [No73]. Thirty senior practitioners were also interviewed to gather their conceptualization of maturity along these four dimensions. Subsequently, based on the interviews and literature reviews, draft instruments were developed. All four instruments were then pretested in small scale studies to ensure consistency in their interpretations. The reliabilities of the instruments used to measure these four dimensions were all quite high (see later in this chapter for the exact results).

Two dimensions of RE success were measured [EM95b]: the quality of RE service, and the quality of RE products. The quality of RE service has two subdimensions: (a) user satisfaction and commitment, and (b) the fit of the recommended solution with the user organization. The quality of RE products has two subdimensions: (a) the quality of the process and data models, and (b) the quality of the cost/benefits analysis.

In order to identify and measure the two dimensions of RE success, the RE literature was reviewed by the authors to determine how RE success had been conceptualized. Thirty senior practitioners were subsequently interviewed to elicit their conceptualizations of RE success and to compare them with the literature derived ones. Based on the interviews 34 criteria for evaluating RE success were developed. Ten senior practitioners were then asked to categorize these criteria and to prioritize them. Using the categorization and prioritization information three dimensions of RE success were identified. Only the most important criteria for each dimension were retained. Eighteen senior practitioners were then interviewed to prioritize the dimensions of RE success. This led the authors to retain the above two dimensions of RE success. These studies provided information to construct a questionnaire for evaluating RE success. This questionnaire was pretested to increase confidence that different people will interpret the questions in a similar manner. This instrument also had high reliability (see [EM95b]).

Data were collected from 38 MIS organizations. Of these 58% were located in Canada, 31.6% in the U.S.A., and 10.5% in Australia. Many of the MIS departments were in large government organizations (greater than CA$1 billion in budget/revenue). All of the data were collected using questionnaires. For measuring RE success, each questionnaire was designated as either a "successful or highly successful" project or "not successful" project. The respondents were left to select a project that they had worked on that also matched the designation. This ensured a reasonable amount of variation in RE success.

2.2.2 Results

The Pearson correlations between the four dimensions of maturity and the quality of RE service dimension are shown in Figure 6. The relationship between the organization dimension and the quality of service is moderate and is statistically significant. This is consistent with the other findings in this chapter.

The relationships with the quality of RE products were all small and not statistically significant. This indicates that perhaps other factors are also important in determining product quality. For instance, one study found that the capabilities of the lead architect were related to the quality of RE products [EM94].

To further understand the relationship between organizational maturity and these outcome variables, conditional relationships were investigated. One possible moderating variable is the size of the MIS organization. To investigate this possibility, the sample of MIS organizations was divided into those that

were small (less than 100 employees) and those that were large (100 or more employees). Then the correlations between maturity and RE success were compared for the small and large MIS organizations[6]. It was found that there are no differences in the correlations between small and large organizations for all the dimensions of maturity. Therefore, MIS organization size does not seem to moderate the relationship. This result is consistent with that found in [GH95].

Measure of Maturity	Quality of Requirements Engineering Service
Standardization	0.26
Project Management	0.22
Tools	0.15
Organization	0.58*

Figure 6: Relationship between maturity dimensions and RE success (* indicates significance at an alpha level of 0.05).

Another possible moderating variable is business sector of the organization. A common differentiation is between government and non-government organizations. Again the organizations were divided, but this time depending on whether they were government or not. Then the correlations between maturity and RE success were compared for these two groups. It was found that there are no large differences in the correlations between government and non-government organizations for all the dimensions of maturity. Although, on the Project Management dimension, the difference does approach statistical significance, indicating that potentially the relationship between the Project Management maturity dimension and the quality of RE service is larger for government organizations. Thus, business sector *may have* a small moderating effect on the maturity $\leftrightarrow$ RE success relationship.

2.3 Does Software Process Improve Project Results?

2.3.1 The Study

Results from a survey conducted at the 1993 Software Engineering Process Group (SEPG) National Meeting in Palo Alto, California [D+95][D+96] were among the first software engineering studies to provide empirical evidence linking the management processes followed by software development projects to differences in their subsequent performance. The survey asked a varied group of developers about their experiences in specific software projects on which they had recently worked.

The questions addressed several aspects of project performance. As seen in Figure 7, three questions about software quality broadly construed were asked: (a) the match between system capabilities and user requirements, (b) ease of use of the delivered software, and (c) the extent of rework that became necessary after the product was delivered. An additional pair of questions were also asked about the projects' success in meeting budget and schedule targets.

[6] To test the significance of the difference in the correlation coefficients, the Fisher transformed z value was used to compare independent r's (see [CC83]).

Software Quality

- Match between system capabilities and user requirements: “Capabilities of the system fit well with customer or user needs.”
- Ease of use; “Customers or users found the system difficult to use.”
- Extent of rework: “major rework was required.”

Meeting Targets

- Within budget: “Project costs exceeded budget.”
- On schedule: “Completion of products or modules was later than scheduled.”

Figure 7: Project performance.

The survey also asked several questions about variations in adherence to software processes that were mapped with varying fidelity to key process areas and common features of the Capability Maturity Model for Software (CMM[sm]). As seen in Figure 8, these include aspects of software project planning, training in the software processes followed by the projects, software process stability, coordination with users or customers, the use of design reviews, use of prototyping, and variations in reliance on cross functional teams.

As seen more fully elsewhere [D+95][D+96], approximately 80 percent of the self-administered questionnaires that were distributed were collected. These are generally experienced software practitioners. Half of them had at least 15 years of experience in the software industry. One-fourth of them had worked in software for 23 years or more.

The analyses reported here are limited to individuals who reported working on projects that had delivered completed products. We asked each respondent to report about his or her experience on the one such project with which s/he was most familiar.

The projects themselves tend to be of substantial size and complexity. Half of them ran for at least 30 months; one-fourth lasted 60 months or more. The median staff size was 23 people; the upper quartile was 46 people. The projects' parent organizations typically are large in size, with a median of 200 software employees and an upper quartile of 550. Over half (55%) of the organizations come from the defense industry, but almost half are drawn from non-defense organizations.

Software Project Planning

- "The project plan and estimates were realistic."
- Project risks were effectively addressed."

Software Process Training

- "Project team members received training in following the software process."

Software Process Stability (integrated software management)

- "The software process followed on this project was similar to that on other projects in your organization."
- "The programming languages and tools used on this project were similar to the ones used on other projects in your organization."

Coordination with users or customers (intergroup coordination)

- "The technical staff was in frequent contact with eventual users of the software."

Design Reviews (peer reviews)

- "A senior team thoroughly reviewed the design."

Prototyping

- "Prototypes of key modules were built before requirements were frozen."

Cross Functional Teams

- "People with relevant technical expertise contributed to requirements analysis."
- "People with relevant application expertise contributed to design and coding."

Figure 8: Software Processes.

As seen in Figure 9, there is evidence that the respondents did in fact try to give honest and forthright answers to the factual questions that were asked. Of course, people who work on software process improvement (SPI) and attend SEPG meetings are even more likely than most others to know the "right answers" to questions about software process. Hence the amount of time that respondents spent working on process improvement was controlled statistically. The overall results were not affected [D+95][D+96]. However those who worked most on SPI actually were somewhat less likely to attribute the quality of their projects' work to good planning. They do tend to report high-quality outcomes, but they are less likely to claim that their projects followed good planning practices. One might conjecture that those more committed to SPI are also more circumspect and have higher standards about what constitutes good process.

They reasonably often admit that they don't know the answers. For example:

- 10% don't know whether the customers found the software difficult to use
- 8% don't know if a senior team did a thorough design review
- 8% don't know if major rework was required
- 10% don't know if project costs exceeded budget

They often admit to "socially undesirable" answers. For example:

- 64% say that project team members were not trained in following the software process
- 37% say a senior team did not do a thorough design review
- 52% say that risks were not effectively addressed
- 49% say that major rework was required
- 65% say that costs exceeded budget
- 77% say that products were completed later than scheduled
- 48% say that the project plan and estimates were unrealistic

Figure 9: Are the respondents truthful?

2.3.2 Results

Consistently following software processes does appear to pay off in better quality software delivered on time and budget. The use of cross functional teams and explicit project planning have the most effect overall in the data presented here. Other measures are better predictors for some, but not all, of the five outcome measures.

One table follows for each outcome measure. This table summarizes the overall effect of the process measures that theoretically should be related to the outcome measure.

The summary tables are based on log likelihood analyses for ordered categorical data [Ev92]. Analogous to analyses of variance, the "U" statistics summarizes "degree of fit" between the process measure(s) and each outcome measure. U statistics tend to be low in magnitude with poorly distributed and non-monotonic survey data. However the overall U statistics generally are higher than those for the individual relationships since the error in the individual relationships is "averaged out." The process measures in the overall models are limited to bivariate relationships that are at least marginally significant ($p < 0.10$) and/or to the five best fitting predictors.

System Capabilities and User Requirements: As seen in Figure 10, the respondents who say that the capabilities of their systems fit well with their customers or users' needs report better use of cross functional teaming and project planning than do those who claim less success in matching system capabilities to user requirements. Most feel that their projects do at least moderately well in meeting their system requirements, and in ensuring that sufficient technical expertise is brought to bear during requirements analysis. Still, the percentage differences are quite substantial. Over 85 percent of those who agree without reservation that people with relevant technical expertise contributed to requirements analysis also agree without reservation that there is a good match between system capabilities and

customer requirements. The comparable percentages are much different for those who have reservations about their projects' performance on either dimension.

	U	P
Overall fit	0.49	0.04
Cross functional teams		
technical expertise: requirements	0.19	0.01
application expertise: design and coding	0.15	0.01
Software project planning		
realistic project plan and estimates	0.15	0.01
project risks effectively addressed	0.13	0.02
Design review by senior team	0.11	0.07

Figure 10: Match between system capabilities and user requirements.

Ease of Use: The authors were somewhat less successful in explaining ease of use. As seen in Figure 11, the overall fit is lower than in Figure 10. Once again, though, cross functional teaming and project planning are the best predictors, along with training in the processes that were followed by these projects. For example, fully 90 percent of those who agree without reservation that risks were effectively addressed in their projects also claim that their customers or users did not find their systems difficult to use. However, over 60 percent of those who said their projects did least well in addressing risk also said that their customers in fact had difficulty with system use.

	U	P
Overall fit	0.33	0.01
Cross functional teams		
technical expertise: requirements	0.12	0.01
application expertise: design and coding	0.08	0.06
Software project planning		
realistic project plan and estimates	0.06	0.17
project risks effectively addressed	0.11	0.01
Process training received	0.07	0.14

Figure 11: Ease of use.

Extent of Rework: As seen in Figure 12, better use of cross functional teaming and project planning, particularly at the requirements stage, appear to pay off in less need for extensive rework. So too does training in the software processes that were followed in the projects about which the respondents reported. Many of the respondents report that their project team members received little if any training, and those who received the least training also said that major rework was in fact required on the projects about which they reported. However, almost 70 percent of those who said that training was done well on their projects said that major rework was not a problem for them.

	U	P
Overall fit	0.59	0.01
Cross functional teams		
technical expertise: requirements	0.12	0.01
application expertise: design and coding	0.07	0.11
Software project planning		
realistic project plan and estimates	0.07	0.11
project risks effectively addressed	0.12	0.01
Process training received	0.10	0.02
Prototypes before requirements frozen	0.07	0.07

Figure 12: Extent of rework.

Meeting Budget: Once again, cross functional teaming and project planning are among the best predictors. Notice in Figure 13 that good coordination between users and the technical staff also appears to pay off in better success in meeting budget targets. So too does consistent training. For example, differences in reported success at planning and estimation are closely related to whether or not the respondents report having their costs exceed their project budgets.

	U	P
Overall fit	0.65	0.01
Software project planning		
realistic project plan and estimates	0.29	0.01
project risks effectively addressed	0.18	0.01
Frequent contact with users	0.11	0.01
Cross functional teams		
technical expertise: requirements	0.09	0.05
application expertise: design and coding	0.09	0.06
Process training received	0.08	0.05

Figure 13: Meeting budget.

Meeting Schedule: Perhaps not surprisingly, the authors were least successful in explaining variation in the respondents' reported ability to meet schedule commitments (Figure 14). Once again, though, project planning, cross functional teaming, and consistent training are the best predictors available. However, almost 80 percent of those who said their projects did a poor job of addressing risk also admit that completion of their products or modules was later than scheduled.

	U	P
Overall fit	0.26	0.01
Software project planning		
realistic project plan and estimates	0.14	0.01
project risks effectively addressed	0.15	0.01
Cross functional teams		
technical expertise: requirements	0.11	0.01
application expertise: design and coding	0.11	0.01
Process training received	0.07	0.10

Figure 14: Meeting schedule.

2.4 User Participation in the Requirements Engineering Process

2.4.1 The Study

This study was a survey of 39 requirements engineering processes in different MIS organizations. The objective was to investigate the effect of user participation on the success of the requirements engineering process. It was hypothesized that uncertainty would moderate the relationship between user participation and RE success. A theoretical model was developed and the following predictions were formally tested:

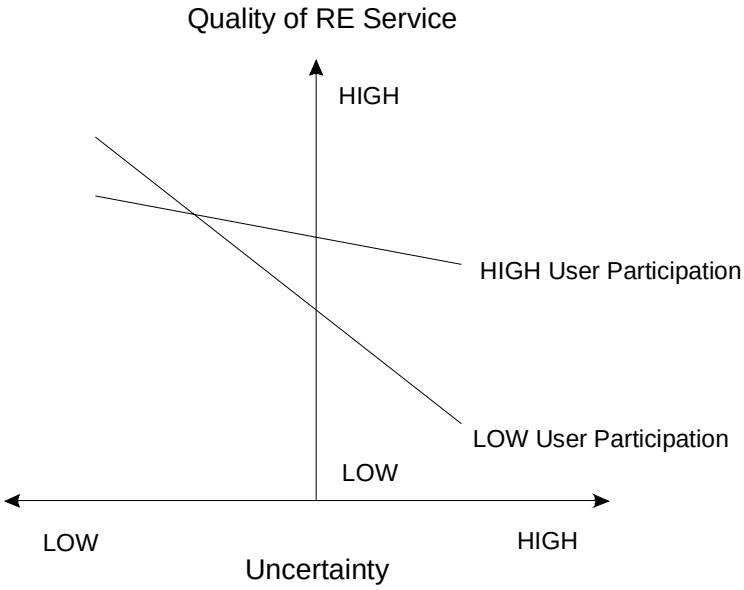
P1 Uncertainty moderates the effect of user participation on RE success. This means that increases in user participation become more effective (in terms of RE success) as uncertainty increases.

P2 User participation buffers the effect of uncertainty on RE success. This means that as uncertainty increases, greater user participation reduces the negative consequences of high uncertainty on RE success.

Instruments for measuring each of the three variables (user participation, uncertainty, and requirements engineering success) in the above predictions were developed. Two dimensions of requirements engineering success were measured as discussed earlier in this chapter. The data analysis utilized multiple ordinary least squares regression with an interaction term (see [JTW90]).

2.4.2 Results

The results of this study are summarized in Figure 15. In terms of the predictions P1 and P2, the quality of RE service model matches them. There is a moderating effect of uncertainty on the user participation and quality of RE service relationship. The slope increases positively with increases in uncertainty as predicted from P1. There is also a buffering effect of user participation on the uncertainty and quality of RE service relationship. The slope decreases with increases in user participation as predicted from P2.



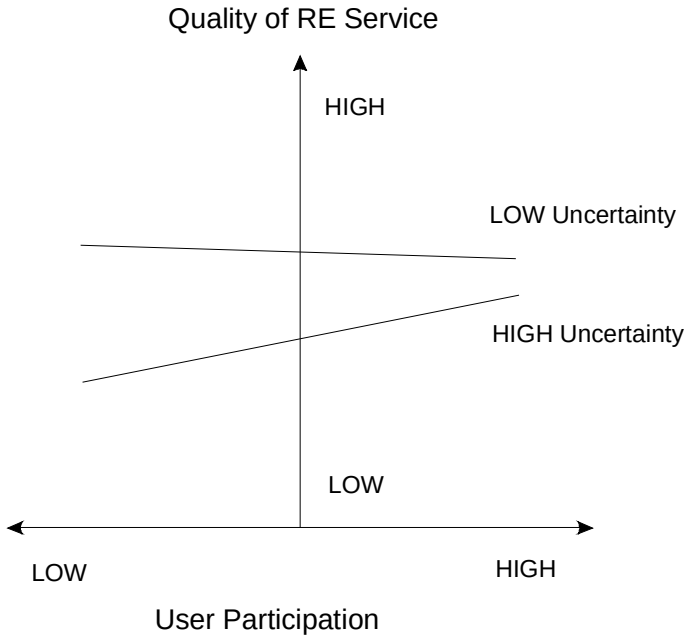


Figure 15: The relationship among user participation, uncertainty, and requirements engineering success.

It was therefore found that the interaction between user participation and uncertainty has a significant impact on the quality of RE service. As uncertainty increases, the importance of user participation also increases. Therefore, greater user participation seems to be a good strategy for alleviating the negative consequences of uncertainty on the quality of RE service. Increased user participation seems to be conducive towards greater user consensus and also it helps them reason about what their business processes should be like and what they want the Information System to do. Furthermore, as uncertainty decreases, the importance of user participation decreases. When uncertainty is low, changes in user participation have no impact on the quality of RE service. Therefore, added participation has little benefit. This does not mean that no participation is necessary, only that increases in participation do not bring about added benefits. Furthermore, it may be that when there is low uncertainty, users resent increases in participation when they feel that it does not contribute substantially. This resentment may bring about reductions in quality of service as user participation increases.

The results of this study indicate that the magnitude of the benefits from some of the practices that are included in contemporary maturity models may be contingent upon project characteristics. In this case, the project characteristic was uncertainty.

2.5 Other Studies

A few other studies have investigated the benefits of organizational maturity. These are summarized briefly below. We also summarize the results of studies that have investigated the benefits of ISO 9000 registration.

2.5.1 Benefits of Organizational Maturity

One correlational study that investigated the benefits of moving up the maturity levels of the CMM[sm] was conducted by Lawlis et al. [LFT95]. They obtained data from historic U.S. Air Force contracts. Two measures were considered: (a) cost performance index which evaluates deviations in actual vs. planned project cost, and (b) schedule performance index which evaluates the extent to which schedule has been over/under-run. results from Levels 1 to 3 only indicate that:

- generally, higher maturity projects approach on-target cost
- generally, higher maturity projects approach on target schedule

In another study, Brodman and Johnson [BJ95] present data that show the benefits of software process improvement based on the CMM[sm]. Their data indicate that some organizations have witnessed increased productivity, reduced defect levels, reduced rework effort, reduction in costs and greater within estimate project completions. However, it is not clear from their data over what period of time these improvements materialized, and also what changes in CMM[sm] levels were associated with these improvements.

Jones [Jo96] presents the results of an analysis on the benefits of moving up the 7-level maturity scale of Software Productivity Research (SPR) Inc.'s proprietary model. This data were collected from SPR's clients. His results indicate that as organizations move from Level 0 to Level 6 on the model they witness (compound totals):

- 350% increase in productivity
- 90% reduction in defects
- 70% reduction in schedules

Since it is difficult to find low maturity organizations with objective data on effort and defect levels, and since there are few high maturity organizations, Jones' data relies on the reconstruction of, at least, effort data from memory, as noted in [Jo94]: "The SPR approach is to ask the project team to reconstruct the

missing elements from memory." The rationale for that is stated as "the alternative is to have null data for many important topics, and that would be far worse." The general approach is to show staff a set of standard activities, and then ask them questions such as which ones they used and whether they put in any unpaid overtime during the performance of these activities. For defect levels, the general approach is to do a matching between companies that do not measure their defects with similar companies that do measure, and then extrapolate for those that don't measure. It should be noted that SPR does have a large data base of project and organizational data, which makes this kind of matching defensible.

2.5.2 Benefits of ISO 9001 Registration

Many software organizations are being assessed against the clauses of ISO 9001. A number of surveys have been conducted that evaluate the benefits of ISO 9001 registration in industry in general and in software organizations in particular. Some of the results of these surveys have been presented in [SPQ94]. Below we summarize some of the relevant findings:

- One survey conducted in 1993 had 292 responses with almost 80% of the responding organizations being registered to ISO 9001. The findings included:
    - 74% felt that the benefits of registration outweighed the costs
    - 54% received favourable feedback from their customers after registration
- A survey of companies in the U.K. had 340 responses from companies that were registered. It was found that 75% of the respondents felt that registration to Iso 9001 improved their product and/or service.
- A survey of companies that were registered in the U.S.A. and Canada with 620 responses found that:
    - the most improtant internal benefits to the organization included: better documentation (32.4%), greater quality awareness (25.6%), a positive cultural change (15%), and increased operational efficiency/productivity (9%); and
    - the most important external benefits to the organization included: higher perceived quallity (33.5%), improved customer satisfaction (26.6%), gaining a competitive edge (21.5%), and reduced customer quality audits (8.5%).
- A survey of 45 software organizations in Europe and North America that have become ISO 9001 registered found that:
    - 26% reported maximum benefit from increased efficiency
    - 23% reported maximum benefit from increased product reliability
    - 22% reported maximum benefit from improved marketing activity
    - 14% reported maximum benefit from cost savings, and
    - 6% reported maximum benefit from increases exports

Thus, with respect to registration to ISO 9001, the few studies that have been conducted are consistent in their findings of benefits to registration. However, many of these studies were not specific to software organizations. Therefore, more research specifically with software organizations would help the community better understand the effects of registration.

3. Reliability

Reliability is an enduring concern for software process assessments. The investment of time, money, and personal effort needed for assessments and successful software process improvement is quite non-trivial, and decisions based on assessment results are often far-reaching. Organizations and acquirers of software systems must be confident that the assessment results are well-founded and repeatable.

Reliability is defined as the extent to which the same measurement procedure will yield the same results on repeated trials and is concerned with random measurement error [CZ79]. This means that if one were to repeat the measurement under similar or compatible conditions the same outcomes would emerge.

There has been a concern with the reliability of assessments. For example, Card discusses the reliability of Software Capability Evaluations in a recent article [Ca92], where he commented on the inconsistencies of the results obtained from assessments of the same organization by different teams. Mention is also made of reliability in a contract award situation where emphasis is placed on having one team assess different contractors to ensure consistency [Ru93]. Bollinger and McGowan [BM91] criticize the extent to which the scoring scheme used in the SEI's Software Capability Evaluations contributes towards reduced reliability (see also [HC91]). The Interim Profile method of the SEI [W+94] includes specific indicators to evaluate reliability. Furthermore, a deep concern with reliability is reflected in the empirical trials of the prospective SPICE standard whereby the evaluation of the reliability of SPICE-conformant assessments is an important focus of study [EG95].

Unreliability in software process assessments is caused by random measurement error. Some common sources of random measurement are presented in Figure 16 [EG95].

Source of Error	Description
Different Occasions	Assessment scores may differ across time. Instability of assessment scores may be due to temporary circumstances and/or actual process changes.
Different Assessors	Assessment scores may differ across assessors (or assessment teams). Lack of repeatability of assessment scores may be due to the subjectivity in the evaluations and judgment of particular assessors (i.e., do different assessors make the same judgments about an organization's processes?).
Different Instrument Contents	Assessment scores may differ across instruments. Lack of equivalence of instruments may be due to the questions in different instruments not being constructed according to the same content specifications (i.e., do different instruments have questions that cover the same content domain?).
Within Instrument Contents	Responses to different questions or subsets of questions within the same instrument may differ amongst themselves. One reason for these differences is that questions or subsets of questions may not have been constructed to the same or to consistent content specifications. Regardless of their content, questions may be formulated poorly, may be difficult to understand, may not be interpreted consistently, etc.

Figure 16: Definition of some sources of error in process assessments.

One important implication of the extent of unreliability is that the score obtained from an assessment is only one of the many possible scores that would be obtained had the organization been repeatedly assessed. This means that, for a given level of confidence that one is willing to tolerate, an assessment score has a specific probability of falling within a range of scores. The size of this range increases as reliability decreases. This is illustrated in Figure 17.

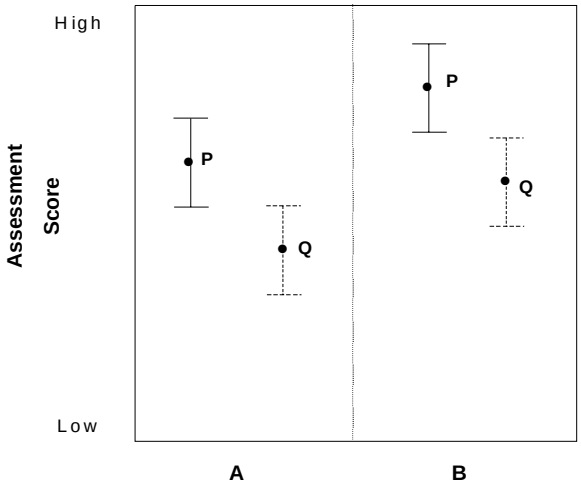


Figure 17: Example hypothetical assessment scores with confidence intervals.

Assume Figure 17 shows the profiles of two organizations, A and B, and that P and Q are two different processes being assessed. Due to random measurement error, the scores obtained for each process are only one of the many possible scores that would be obtained had the organization been repeatedly assessed. While obtained scores for organization B are in general higher than those of organization A, this may be an artifact of chance. Without consideration of random measurement error, organization A may be unfairly penalized in a contract award situation. Turning to a self-improvement scenario, assume that Figure 17 shows the profiles of one organization at two points in time, A and B. At time A, it may seem that the score for process Q is much lower than for process P. Thus, the organization would be tempted to pour resources on improvements in process Q. However, without consideration of random measurement error, one cannot have high confidence about the extent to which the difference between P and Q scores is an artifact of chance. Furthermore, at time B, it may seem that the organization has improved. However, without consideration of random measurement error, one cannot have high confidence about the extent to which the difference between A and B scores (for processes P and Q) are artifacts of chance.

The above examples highlight the importance of evaluating the extent of reliability of software process assessments. In this section we present three studies that have employed three different methods for evaluating the reliability of software process assessment methods.

3.1 Assessor and Assessee Perceptions

One of the most straight forward ways to evaluate the reliability of assessments is to ask individuals who were involved with assessments about their perceptions of assessment repeatability and consistency. This was one of the approaches that was followed during the field trials in the SPICE project[7].

A set of questions on the repeatability and consistency of assessments were administered during the SPICE trials. Responses were obtained from both, assessors and assessees who took part in the assessments. In total, data from 35 assessments were collected before the response deadline. Of these, 20 were conducted in Europe, 1 in Canada, and 14 in the Pacific Rim.

The general approach to the analysis of the questionnaire data were to identify the proportions of respondents who are supportive (as opposed to critical) of SPICE and its various elements. A supportive response is one:

- that says something positive about SPICE, and/or
- that will not require any changes to the draft SPICE documents (i.e., the ones that were used during the trials assessments)

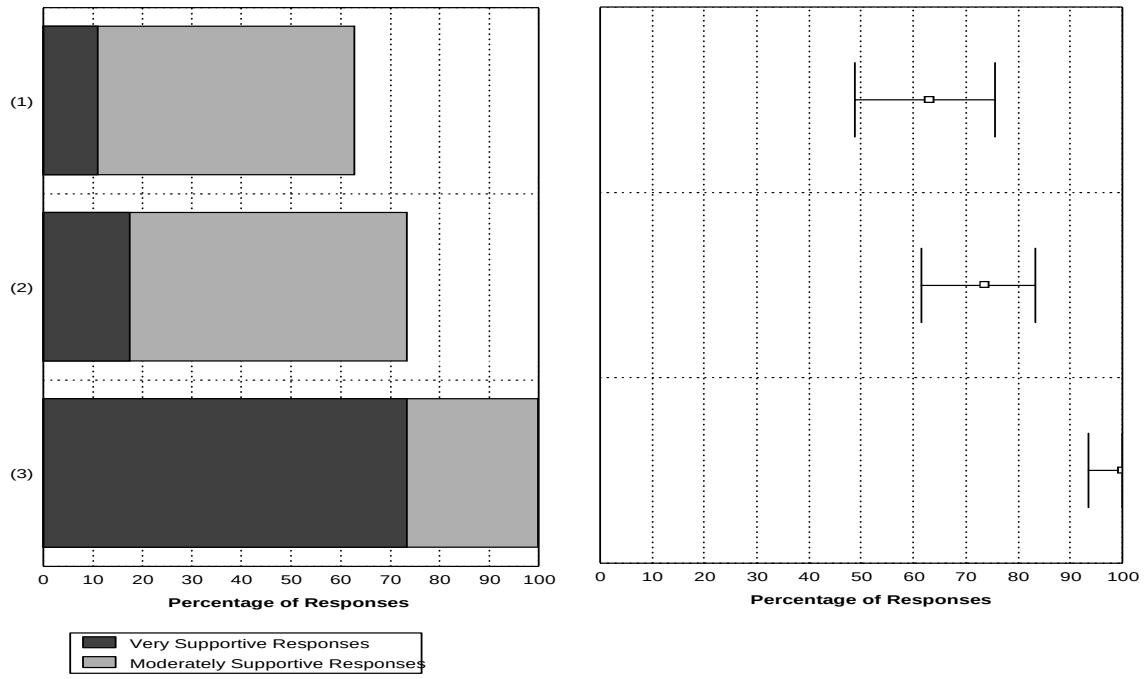
The complete results of the phase 1 SPICE trials are reported in [SP95]. Here, we show some of the results of the questions that concerned the repeatability and consistency of SPICE assessments [EG96b]. The results on the perceptions of repeatability and consistency were considered by the SPICE project in revising the draft SPICE documents.

For the supportive responses, a distinction between “very supportive” and “moderately supportive” responses was made. This distinction helps make clear the extent of support for SPICE. The correspondence between these responses and the response categories in the qustionnaires is given in Figure 18. There are two kinds of chart: a bar chart and a range plot. The bar chart is straightforward to understand. The range plot shows the 80% confidence interval for the proportion of respondents supporting SPICE.

---

[7] The SPICE Project’s commitment to conduct empirical trials is unique among international software engineering standards efforts. From the beginning, project members have recognized the need for objective evidence to back up, and inform, their assertions.

The trials are scheduled to be completed in three broad phases. The first phase was completed in calendar year 1995. Its results are based on several sources of data, including a series of questionnaires completed by both assessors and assessees fro over 30 assessments conducted world-wide, project problem reports and change requests, and the actual rating profiles forthcoming from the assessment. The focus of phase 1 was on the usability and clarity of the SPICE document set. The results were used to help identify shortcomings and inform decisions about the content of the document set for resolution prior to standardization. More details on the SPICE trials and their outcomes may be found in [EG95][EG96][MO95][WH96][M+96].



No.	Question	Supportive Response Categories	Critical Response Categories	Percentage Supportive
(1)	The assessment results were too dependent on the expertise and judgement of the assessment team (assessees' responses)	• Strongly Disagree • Disagree	• Strongly Agree • Agree	(17/27) = 63%
(2)	The assessment results were too dependent on the exprtise and judgement of the assessment team (assessors' responses)	• Strongly Disagree • Disagree	• Strongly Agree • Agree	(25/34) = 73%
(3)	In general, how difficult was it for the assessment team to come to consensus? (assessors' responses)	• Not Very Difficult • Moderately Difficult	• Extremely Difficult • Very Difficult	(34/34) = 100%

Figure 18: General impressions of repeatability of assessments.

One measure of repeatability is the extent of difficulty that the assessors had in coming to consensus about their rating decisions. Although it is an indirect measure at best, the experienced assessors were asked to characterize the difficulty their assessment teams had in coming to consensus. As seen in Figure 18, *all* of them reported that they reached consensus with moderate difficulty at worst; over 70 percent said it was not very difficult at all.

Of course, it is possible that this lack of difficulty in coming to consensus is due to the relative experience of the phase 1 assessors. Indeed, almost 30 percent of the assessors, and almost 40 percent of the assessment sponsors, agree that the assessment results were too dependent on the expertise and judgment of the phase 1 assessment teams (Figure 18). Fewer than 20 percent of either group disagrees strongly.

3.2 Inter-rater Agreement Studies

Measurement error due to different assessors (or assessment teams) probably accounts for a large proportion of the error in software process assessments. Inter-rater agreement studies would account for this source of error. The basic idea behind inter-rater agreement studies is to have an organization assessed by more than one team and then compare the extent to which the ratings from the different assessment teams agree. More detailed guidelines for conducting inter-rater agreement studies are given in Figure 19.

During the SPICE trials, a number of inter-rater agreement studies were conducted [E+96]. These studies should be considered as preliminary since only a relatively small number of data has been collected thus far. However, they do demonstrate the principles of evaluating inter-rater agreement, and did produce interesting results worthy of further investigation.

Instructions for Conducting Inter-Rater Agreement Studies

- Divide the assessment team into two groups with at least one person per group
- The two groups should be selected so that they are as closely matched as possible with respect to training, background, and experience
- The two groups should use the same evidence (e.g., attend the same interviews, inspect the same documents, etc.), method, and tools
- The two groups independently rate the same subset of primitive process instances
- After the independent ratings, the two groups then meet to harmonize their ratings for the final SPICE profile
- There should be no discussion between the two groups about rating judgement prior to harmonization, and the first group examining any physical artifacts should leave them as close as possible (organized/marked/sorted) to the state that the assessees delivered them.

Figure 19: Guidelines for conducting inter-rater agreement studies.

A commonly used statistic for evaluating inter-rater agreement is the proportion of agreement [Fle81] (i.e., the proportion of ratings for which there was 100% agreement amongst the different assessment teams). However, this statistic includes agreement that could have occured by chance. For example, if there were two teams and they employed completely different criteria for assigning their ratings to the

same organizational practices, then a considerable amount of observed agreement would still be expected by chance. A statistic that takes into account agreement that has been obtained by chance is Cohen's kappa (κ) [Coh60].

If there is complete agreement, $\kappa=1$. If observed agreement is greater than chance, then $\kappa>0$. If observed agreement is less than chance, then $\kappa<0$. The minimum value of κ depends upon the marginal proportions. However, since we are interested in evaluating agreement, the lower limit of κ is not of interest.

After calculating the value of Kappa, the next question is "how does one interppret it?" Landis and Koch [LK77] have presented a table that has been found to be useful for benchmarking the obtained values of Kappa. This is shown in Figure 20.

Kappa Statistic	Strength of Agreement
<0.00	Poor
0.00-0.20	Slight
0.21-0.40	Fair
0.41-0.60	Moderate
0.61-0.80	Substantial
0.81-1.00	Almost Perfect

Figure 20: The interpretation of values of kappa.

Figure 21 shows the results from two different assessments that were conduced during the SPICE trials. In both cases the number of independent teams was 2. For the first assessment the "Develop Software Design" process (ENG.3) was assessed. For the second assessment, which was conducted in a different organization, the "Manage Quality" process (PRO.5) was assessed. The figure shows the proportion of agreement between the two teams, the kappa coefficient, and its interpretation.

From this limited data, it is evident that, in general, the two separate teams tended to agree quite highly. These results can be considered encouraging for SPICE. However, further analysis reported in [E+96] identified differences in the extent of agreement between low and high capability level ratings (there tended to be less agreement on high capability level ratings). This is thus an example of how inter-rater agreement studies can be used to evaluate repetability and also to investigate where there may be potential problems.

	Proportion Agreement	Kappa	Interpretation
Study 1 (ENG.3 Process)	0.77	0.59	Moderate
Study 2 (PRO.5 Process)	0.78	0.70	Substantial

Figure 21: Results of inter-rater agreement studies in SPICE.

3.3 Internal Consistency Methods

Estimates of the reliability of software process assessments using an internal consistency method account for "within instrument contents" as a source of error (see Figure 16). One of the most commonly used internal-consistency estimates is called Cronbach alpha, named after the person whose paper popularized it [Cr51]. In the related discipline of Management Information Systems (MIS), researchers developing instruments for measuring software processes and their outcomes tend to report the Cronbach alpha coefficient most frequently [EG95]. Furthermore, some MIS researchers consider the Cronbach alpha coefficient to be the most important reliability estimate [SK91].

The Cronbach alpha coefficient is a number that ranges from 0 to 1. A value of 0 indicates that the measure has no reliability whatsoever (i.e., all variation is due to random error). A value of 1 indicates that the measure has perfect reliability. Of course it is desirable that the alpha value approaches 1. In general, an alpha value of 0.9 is considered sufficiently large for practical decision making situations, and a minimal value of 0.8 is considered to be sufficient for research purposes [Nu78].

To our knowledge, there are only two reports in the literature on the results of evaluating the reliability of software process assessments using the internal consistency method. The first study was mentioned by Humphrey and Curtis [HC91]. In that article, they report the Cronbach alpha coefficients calculated from level 2 and 3 questions on an earlier version of the SEI's maturity questionnaire to be quite high (0.9). They, however, do not give the details of that study. The second study was conducted by El Emam and Madhavji [EM95a]. Brief details of the latter study are given below.

El Emam and Madhavji developed a maturity assessment instrument that can be used for assessing MIS organizations. In their paper, they consider the following dimensions of maturity: (a) standardization, which is concerned with with process and product standardization in the MIS organization, (b) project management, which is concerned with the extent to which good project management practices are employed, (c) tools, which is concerned with effective automated tool usage in the organization, and (d) organization, which is concerned mainly with the documentation of the overall organization's missions and goals and the elignment of the MIS organization with these.

External consultants assessed 38 MIS organizations world-wide. Of these, the majority (58%) were located in Canada. Also, many of the surveyed MIS departments were in large government organizations (greater than CA$1 billion in budget/revenue).

The results of the reliability estimates using the Cronbach alpha coefficient are shown in Figure 22. As can be seen, each of the four maturity dimensions has relatively high reliability according to the guidelines mentioned above. Furthermore, these are comparable to the 0.9 value obtained by Humphrey and Curtis [HC91].

Maturity Dimension	**Cronbach Alpha Coefficient**
Standardization	0.88
Project Management	0.91
Tools	0.88
Organization	0.82

Figure 22: Reliability estimates for the four dimensions of maturity.

4. Conclusions: What does it all mean?

4.1 Validity

More empirical evidence already exists than is sometimes realized. And our understanding of the effects of maturity is starting to improve. First of all, in general, maturity does appear to matter. By now we have found some noticeably different patterns of quality, productivity, and/or predictability as a function of differences in software process in a variety of studies (section 2).

Of course, not all improvement efforts will be equally successful. Neither will all assessment results be equally accurate. But the overall general conclusion remains. Attention to software process can pay off in better performance. Similar results exist for smaller as well as large software organizations, both in defense and elsewhere in the software industry, but perhaps not so similarly between government and non-government organizations (sections 2.1 and 2.2).

We cannot expect process maturity to explain everything. Perfect, one-to-one, relationships between process maturity and measures of performance or product will rarely if ever exist. Other mediating factors, both technical and people related, undoubtedly are important as well.

Neither can we expect all aspects of process maturity to explain the same performance dimensions equally well. Processes that improve product quality will not necessarily result in increased productivity or schedule predictability. Similarly, processes aimed at improving requirements engineering will differ from those for testing, design, or reengineering. We need to examine finer grained distinctions to guide process improvement aimed at more specific aspects of organizational performance.

4.2 Reliability

The bottom line is how confident can we be that assessment results are repeatable? Would we expect comparable results if the same organization was assessed by different assessment methods or different assessors? Much more remains to be learned before we can fully answer, or even ask, such questions. However the results of the empirical studies done to date do provide some confidence that our current measures of process maturity and capability are reasonably well founded (section 3).

Still, process assessments remain imperfect. There is unreliability associated with any measure, and process assessments are no exception. After all, they do rely on the exercise of human judgment. It is easy to over interpret the meaning of assessment results, especially when making relatively fine distinctions among organizations or across time. The stakes are often high, especially when comparing among competing suppliers for large contracts. Due caution should be exercised. Assessments can inform our best judgment, not replace it.

4.3 What's Next

Validity: Based on the empirical evidence reviewed in this chapter, one can conclude that process maturity is generally associated with better performance in software organizations. The results do vary, but they thus provide direction for further research. For example:

- Maturity is a high level, abstract concept. Different aspects of maturity may have very different effects on project and organizational performance. When combined into one summary dimension, the effects of individual aspects of process maturity may be hidden. More attention must be paid to specific processes and their possibly differing effects.

- Some of the benefits of software engineering practices, as expressed in contemporary process models and methods, may be contingent on organizational and other contextual differences. These contextual effects must be understood in much greater detail.

Reliability: The studies reviewed in this chapter represent much of the published research that examines the reliability of software process assessments. The number of studies of this particular topic is not large, but the cummulative evidence thus far suggests that assessments can in fact be done reliably. However several likely sources of measurement error have not yet been addressed in sufficient depth. In particular we need to know more about the extent to which assessment results are a function of the people who do the assesments, or the assessment methods and tools they use. Such work will be a major emphasis of the second phase of the international SPICE trials. We anticipate doing extensive comparisons of the repeatability of overall results and rating judgments made by independent teams during the same assessments.

Our challenge: A considerable amount of empirical work has already been done that helps us better understand software process improvement and its impact on organizational performance. The work that has been done thus far has followed a variety of research strategies, ranging from uncontrolled case studies to quasi-experimental analyses, but the general results are similar. More researchers are focusing their attention on such topics. However a great deal more remains to be done.

Our challenge, as empirical analysts as well as software practitioners, is to learn more about the conditions where process improvement efforts will work well or poorly. In so doing, we will become better able to translate assessment results into useful guidance for software process improvement.

References

[Ba94] J. Bach: “The Immaturity of the CMM”. In *American Programmer*, 7(9):13-18, September 1994.

[Ba95] J. Bach: “Enough About Process: What we Need Are Heroes.” In *IEEE Software*, 12(2):96-98, February 1995.

[Ba96] R. Basque: “CBA-IPI: How to Build Software Process Improvement Success in the Evaluation Phase?”. In *Software Process Newsletter*, IEEE Computer Society, No. 5, pages 4-6, Winter 1996.

[BF95] S. Benno and D. Frailey: “Software Process Improvement in DSEG: 1989-1995”. In *Texas Instruments Technical Journal*, 12(2):20-28, March-April 1995.

[BM91] T. Bollinger and C. McGowan: “A Critical Look at Software Capability Evaluations”. In *IEEE Software*, pages 25-41, July 1991.

[BJ95] J. Brodman and D. Johnson: "Return on Investment (ROI) from Software Process Improvement as Measured by US Industry". In *Software Process: Improvement and Practice*, 1(1), John Wiley & Sons, 1995.

[Bu95] K. Butler: “The Economic Benefits of Software Process Improvement”. In *Crosstalk*, 8(7):14-17, July 1995.

[Ca92] D. Card: “Capability Evaluations Rated Highly Variable”. In *IEEE Software*, pages 105-106, September 1992.

[CZ79] E. Carmines and R. Zeller: *Reliability and Validity Assessment*, Sage Publications, Beverly Hills, 1979.

[Coh60] J. Cohen: “A Coefficient of Agreement for Nominal Scales”. In *Educational and Psychological Measurement*, XX(1):37-46, 1960.

[CC83] J. Cohen and P. Cohen: *Applied Multiple Regression/Correlation Analysis for the Behavioral Sciences*, Lawrence Erlbaum Associates, 1983.

[Cr51] L. Cronbach: “Coefficient Alpha and the Internal Consistency of Tests”. In Psychometrika, pages 297-334, September 1951.

[Di92] R. Dion: “Elements of a Process Improvement program”. In IEEE Software, 9(4):83-85, July 1992.

[Di93] R. Dion: “Process Improvement and the Corporate Balance Sheet”. In *IEEE Software*, 10(4):28-35, July 1993.

[D+95] C. Deephouse, D. Goldenson, M. Kellner, and T. Mukhopadhyay: “The Effects of Software Processes on Meeting Targets and Quality”. In *Proceedings of the Hawaiian International Conference on Systems Sciences*, vol. 4, pages 710-719, January 1995.

[D+96] C. Deephouse, T. Mukhopadhyay, D. R. Goldenson, and M. I. Kellner: "Software Processes and Project Performance". In *Journal of Management Information Systems*, 12(3):185-203, Winter 1995-96.

[Do93] A. Dorling: “SPICE: Software Process Improvement and Capability dEtermination”. In *Information and Software Technology*, 35(6/7):404-406, June/July 1993.

[Dr94] J-N Drouin: “Software Quality - An International Concern”. In *Software Process, Quality & ISO 9000*, 3(8):1-4, August 1994.

[Dr95] J-N Drouin: “The SPICE Project: An Overview”. In *Software Process Newsletter*, IEEE Computer Society, No. 2, pages 8-9, Winter 1995.

[Dy95] K. Dymond: “Essence and Accidents in SEI-style Assessments or ‘Maybe This Time the Voice of the Engineer will be Heard’”. In *Proceedings of the ISCN-ESI Conference on Practical Improvement of Software Processes and Products*, September 1995.

[E+96] K. El Emam, D.R. Goldenson, L. Briand, and P. Marshall, “Inter-rater Agreement in SPICE Based Assessments: Some Preliminary Results" (in preparation), 1996.

[EG96] K. El Emam and D. R. Goldenson: “Some Initial Results from the International SPICE Trials”. In *Software Process Newsletter*, IEEE Computer Society, No. 6, Spring 1996.

[EG96b] K. El Emam and D. R. Goldenson: "An Empirical Evaluation of the Prospective International SPICE Standard". To appear in *Software Process Improvement and Practice Journal*, John Wiley, 1996.

[EQM96] K. El Emam, S. Quintin, and N. H. Madhavji: "User Participation in the Requirements Engineering Process: An Empirical Study". In *Requirements Engineering Journal*, 1(1), 1996.

[EG95] K. El Emam and D. R. Goldenson: “SPICE: An Empiricist’s Perspective”. In *Proceedings of the Second IEEE International Software Engineering Standards Symposium*, pages 84-97, August 1995.

[EM95a] K. El Emam and N. H. Madhavji: “The Reliability of Measuring Organizational Maturity”. In *Software Process Improvement and Practice Journal*, 1(1):3-25, 1995.

[EM95b] K. El Emam and N. H. Madhavji: "Measuring the Success of Requirements Engineering Processes". In *Proceedings of Second IEEE International Symposium on Requirements Engineering*, pages 204-211, 1995.

[EM94] K. El Emam and N. H. Madhavji: "A Method for Instrumenting Software Evolution Processes and An Example Application". In *Notes From The International Workshop on Software Evolution, Processes, and Measurements*, Technical Report #94-04 NT, Software Engineering Test Lab, Department of Computer Science, University of Idaho, 1994.

[Ev92] B. Everitt: *The Analysis of Contingency Tables*, Chapman & Hall, 1992.

[Fle81] J. Fleiss: *Statistical Methods for Rates and Proportions*, John Wiley & Sons, 1981.

[GH95] D. Goldenson and J. Herbsleb: "After the Appraisal: A Systematic Survey of Process Improvement, its Benefits and Factors that Influence Success". Technical Report, CMU-SEI-95-TR-009, Software Engineering Institute, 1995.

[HZ95] W. Hayes and D. Zubrow: “Moving On Up: Data and Experience Doing CMM-Based Software Process Improvement.” Technical Report, CMU/SEI-95-TR-008, Software Engineering Institute, 1995.

[HG96] J. Herbsleb and D. R. Goldenson: “A Systematic Survey of the Results of CMM-Based Software Process Improvement.” In *Proceedings of the 18th International Conference on Software Engineering*, pages 323-330, 1996.

[HC+94] J. Herbsleb, A. Carleton, J. Rozum, J. Siegel, and D. Zubrow: "Benefits of CMM-Based Software Process Improvement: Initial Results". Technical Report, CMU-SEI-94-TR-13, Software Engineering Institute, 1994.

[Her93] A. Hersh: “Where’s the Return on Process Improvement?”. In *IEEE Software*, page 12, July 1993.

[HSW91] W. Humphrey, T. Snyder, and R. Willis: "Software Process Improvement at Hughes Aircraft". In *IEEE Software*, pages 11-23, July 1991.

[HC91] W. Humphrey and B. Curtis: “Comments on 'A Critical Look'“. In *IEEE Software*, pages 42-46, July 1991.

[JTW90] J. Jaccard, R. Turrisi, and C. Wan: *Interaction Effects in Multiple Regression*, Sage Publications, 1990.

[Jo96] C. Jones: "The Pragmatics of Software Process Improvements". In *Software Process Newsletter*, IEEE Computer Society TCSE, No. 5, pages 1-4, Winter 1996.

[Jo95] C. Jones: “Gaps in SEI Programs.” In *Software Development*, 3(3):41-48, March 1995.

[Jo94] C. Jones: Assessment and Control of Software Risks, Prentice-Hall, 1994.

[Kra94] H. Krasner: "The Payoff for Software Process Improvement (SPI): What it is and How to Get It". In *Software Process Newsletter*, IEEE Computer Society TCSE, No. 1, pages 3-8, September 1994.

[Ker86] F. Kerlinger: *Foundations of Behavioral Research*, Holt, Rinehart, and WInston, 1986.

[Ko94] M. Konrad: “On the Horizon: An International Standard for Software Process Improvement”. In *Software Process Improvement Forum*, pages 6-8, September/October 1994.

[KBD96] P. Kuvaja, A. Bicego, and A. Dorling: "SPICE: The Software Process Assessment Model". In *Proceedings of ESI-ISCN ’95: Practical Improvement of Software Processes and Products*, 1995.

[LK77] J. Landis and G. Koch: “The Measurement of Observer Agreement for Categorical Data”. In *Biometrics*, 33:159-174, March 1977.

[LFT95] P. Lawlis, R. Flowe, and J. Thordahl: "A Correlational Study of the CMM and Software Development Performance". In *Crosstalk*, 8(9):21-25, September 1995.

[Leb96] L. Lebsanft: "Bootstrap: Experiences with Europe's Software Process Assessment and Improvement Method". In *Software Process Newsletter*, IEEE Computer Society, No. 5, pages 6-10, Winter 1996.

[LB92] W. Lipke and K. Butler: “Software Process Improvement: A Success Story”. In *Crosstalk*, 5(9):29-39, September 1992.

[M+96] P. Marshall, F. Maclennan, and M. Tobin: “Analysis of Observation and Problem Reports from Phase 1 of the SPICE Trials”. In *Software Process Newsletter*, IEEE Computer Society, No. 6, pages 10-12, Spring 1996.

[MO95] F. Maclennan and G. Ostrolenk: “The SPICE Trials: Validating the Framework”. In *Proceedings of the 2nd International SPICE Symposium*, Brisbane, June 1995.

[MC96] J. Mayrand and F. Coallier: "System Acquisition Based on Software Product Assessment". In *Proceedings of the 18th International Conference on Software Engineering*, pages 210-219, 1996.

[No73] R. Nolan: "Managing the Computer Resource: A Stage Hypothesis". In *Communications of the ACM*, 16(7):399-405, July 1973.

[Nu78] J. Nunnally: *Psychometric Theory*, McGraw-Hill, 1978.

[PK94] M. Paulk and M. Konrad: “Measuring Process Capability Versus Organizational Process Maturity”. In *Proceedings of the 4th International Conference on Software Quality*, 1994.

[RO95] T. Rout: “SPICE: A Framework for Software Process Assessment”. In *Software Process Improvement and Practice Journal*, Pliot Issue, pages 57-66, August 1995.

[Ru93] D. Rugg: “Using a Capability Evaluation To Select A Contractor”. In *IEEE Software*, pages 36-45, July 1993.

[SK95] H. Saiedian and R. Kuzara: “SEI Capability Maturity Model’s Impact on Contractors”. In *Computer*, 28(1):16-26, January 1995.

[SK91] V. Sethi and W. King: “Construct Measurement in Information Systems Research: An Illustration In Strategic Systems”. In *Decision Sciences*, 22:455-472, 1991.

[SP95] SPICE Project: “SPICE Phase 1 Trials Report”, 1995.

[SPQ94] Staff: “A Survey of the Surveys on the Benefits of ISO 9000”. In *Software Process, Quality & ISO 9000*, 3(11):1-5, November 1994.

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

[WR93] H. Wohlwend and S. Rosenbaum: “Software Improvements in an International Company”. In *Proceedings of the International Conference on Software Engineering*, pages 212-220, 1993.

[WH96] I. Woodman and R. Hunter: “Analysis of Assessment data from Phase 1 of the SPICE Trials”. In *Software Process Newsletter*, IEEE Computer Society, No. 6, pages 5-10, Spring 1996.