

Evaluating Return On Investment For Software Process Improvement Projects

Daniel P. Shoemaker (shoemadp@udmercy.edu) University of Detroit Mercy
Gregory W. Ulferts (ulfertgw@udmercy.edu) University of Detroit Mercy
Patrick T. Wirtz (wirtzpt@udmercy.edu) University of Detroit Mercy
Antonio Drommi (drommia@udmercy.edu) University of Detroit Mercy

Abstract

This article presents a simple approach that will allow decision-makers to evaluate the return on investment of software process improvement prior to launching such an effort. Obviously, it will be easy to tell ten years up the road whether the right decision was made. But a CEO, or CIO contemplating laying out six, or seven figures for the additional personnel and resources to conduct SPI is not in a position to make that call and the wise ones will not be led into it by blind faith. The problem is assessing the risks and returns of such a project in terms and perspective that a non-technical decision-maker can understand. We believe our instrument serves that purpose.

1.0 Introduction: The Problem

Software busts schedules and budgets in a way that would not be tolerated in any other industry. It is a fact that... *Depending on project size, between 25% and 50% of all projects fail, where "failure" means that the project is canceled or grossly exceeds its schedule estimates* (Laker, 1998). A recent Standish Group survey of 8,000 software projects found that the average exceeded its planned budget by 90 percent and its schedule by 120 percent (Construx, 1998). Several industry studies have reported that fewer than half of the software projects initiated in this country finish within their allotted schedules and budgets (Construx, 1998).

This is not a new phenomenon. A study done by the GAO in the 1980s found that fully two-thirds of the software delivered to the federal government was never used and an additional 29% was never delivered at all. The good news was that 3% was usable after changes and 2% could be used as delivered. As a result, the GAO estimated that throughout the 1980s the federal Government's bill for worthless software topped \$150 billion (Quoted in Humphrey, 1994). When 95% of the software delivered to the federal government is worthless you might expect some accountability. Yet numerous studies since then have documented the same problems. These include: 1) *Poor project planning*, 2) *Inadequate documentation of project requirements*, 3) *Insufficient understanding of the business*, 4) *Lack of support and involvement from senior management*, and 5) *No written quality plan or no effective implementation of the plan* (SEI, 1997).

The Standish Group found that the most common causes of project failure were management-based considerations. That covered such things as incomplete requirements, lack of user involvement, lack of resources, unrealistic expectations, lack of executive support, and changing requirements. Those causes occurred with approximately equal frequency (Construx, 1998). A similar study conducted by KPMG Pete Marwick found that 87% of failed projects exceeded their initial schedule estimates by 30% or more. While at the same time 56% exceeded their budget estimates by 30% or more and 45% failed to produce expected benefits. This resulted primarily from the following causes (KPMG, 1997)

Readers with comments or questions are encouraged to contact the authors via email

1. Project objectives not fully specified (51%)
2. Bad planning and estimating (48%)
3. Technology that is new to the organization (45%)
4. Inadequate, or no project management methodology (42%)
5. Insufficient experienced staff on the team (42%)
6. Poor performance by suppliers of hardware/software (42%)

It would be a cop out to suggest that these failures were a consequence of extreme project size, or complexity. In actuality 60% of these failed projects were categorized by KPMG as small. The fact is that small projects (e.g., those that are characteristic of the average mom-and-pop software shop) are almost always over schedule (92%). In fact the larger, more complex projects actually did better. KPMG found that only 86% of these had problems meeting their delivery dates (which is still a pathetic statistic). One reason cited for the success of the big projects was that *formal project and risk management techniques were almost always employed in their management.*

Which leads to the inescapable conclusion that any organization, large or small, simple or complicated, functions better with some sort of defined management structure. The overall purpose of which is to insure that the organization's people equipment and financial resources are utilized efficiently. That requires understanding all of the purposes and intents of the business. The most telling result of the KPMG study was the impact of the general business environment on software project success. Between 44% and 48% of the reasons for project failure came as a consequence of the failure of the software people to clearly understand how the business operated. Exacerbating that problem was the third most common cause of failure, which was the lack of involvement and support from managers. Where projects failed the most common cause was a lack of project management (execution) and either a lack of skill, or an inability to monitor project activity on the part of the project manager (KPMG, 1998).

2.0 Quality Management, Solution or Silver Bullet?

In 1987 Watts Humphrey published an article about assessing software engineering capability (CMU/SEI-87-TR-023, 1987). That was developed into the early Capability Maturity Model described in *Characterizing the Software Process* (Humphrey, 1988) and *Managing the Software Process* (Humphrey, 1989). Version 1.0 of the CMM was released in August of 1991 in two technical reports, *Capability Maturity Model for Software* (Paulk, 1991), and *Key Practices of the Capability Maturity Model* (Weber 1991). This first version would quickly become the Capability Maturity Model (CMM 1.1), which was rolled out in 1993 by Mark Paulk and Bill Curtis (Paulk, 1993). In the meantime, recognizing the limitations of ISO 9000 for software ISO was in the process of developing a much more powerful assessment based certification standard. This went under the informal name of SPICE throughout the 1990s and was formalized as the ISO TR 15504 Standard in May of 1998. The US has finished the second phase of the field trials for this standard and is expected to complete phase three trials in 2000. Promulgation of the ISO 15504 Standard is expected in 2001.

All of these quality system standards concern themselves with the way an organization goes about its work not (directly at least) the outcomes of that work. In other words, they concern themselves with processes, not products under the assumption that if the production and management system is right the product or service that it produces will also be correct. In the case of all of these Standards, the philosophy is that the requirements are generic. Which means that no matter what the organization is or does, if it wants to establish a quality management system, the essential features are spelled out. These are contained in the must address clauses of ISO 9000 or in the key processes and common features of CMM and ISO 15504. It should be noted that quality management frameworks such as CMM, or ISO 9000 provide the organization with a template for setting up and running a quality system.

In concept a quality management system that follows such a defined model, or "conforms to a standard", embodies these state-of-the-art practices. The end-result of this conformance is much improved organizational efficiency and effectiveness. These can be intangible as well as bottom line gains. Brodman (1995) reports on many non-measurable benefits from such practices. These include... *"Improved morale by the developers increased*

respect for software from organizations external to software and less required overtime” (quoted in DACS, 1999). Brodman notes that some organizations looked at benefits from SPI not just in financial terms, but in terms of being more competitive (cheaper and better), improved customer satisfaction (fewer post release problems in the software) and more repeat business from their customers (quoted in DACS, 1999).

Since CMM was introduced a number of reports and papers have been circulated that discuss the costs and benefits of that model. Herbsleb (1994) provided statistical results as reported by 13 organizations to demonstrate the expected value of CMM-based Software Process Improvement. His findings were primarily focused on Level 2 and Level 3 organizations. They show gains in productivity due to... *“Better requirements elicitation, better software management, and incorporation of a software reuse program. Gains in early detection of defects and reductions in calendar time were primarily attributed to reuse... There was no apparent correlation between years of SPI and ROI”* (DACS, 1999). The Boeing Space Transportation Systems (STS) Defense and Space Group reports that improved software processes now find nearly 100% of all defects. Although this increased the design effort by 25% (4% of total development time), it reduced rework during testing by 31% (of total development time). So a 4% increase in effort returned a 31% reduction in rework resulting in a 7.75:1 ROI (Yamamura and Wigle, 1997). Raytheon characterized the benefit of their improvement program by differentiating their costs into the categories of doing it right the first time versus the cost of rework. Based on their process improvement program, Raytheon was able to report that it had eliminated \$15.8 million in rework in less than 5 years (DACS, 1999).

Reports of such glowing success for SPI are all well and good, however the problem with most of these studies is that they have been conducted in organizations that typically work much closer to the leading edge than the average IT firm. Therefore the results tend to be obscured by the fact that the projects on which they are based are not typical of common IT operation. What’s been missing to this point is a simple mechanism that will allow an everyday businessperson to assess the value of formal SPI in their day-to-day practice and that is the objective of the rest of this article.

3.0 ROI as a Decision Factor in Software Process Improvement Projects

This section introduces a technique for evaluating the return on investment in Software Process Improvement (SPI) by comparing its risks against all potential gains. Strassman (1990) believes that risk analysis is a very important aspect of appraisal. According to him, *“Risk analysis is the correct analytical technique with which one can examine the uncertainty of Information Technology investments prior to implementation.”* He believes that *“By making the risks of technology more explicit, you create a framework for diagnosing, understanding and containing the inherent difficulties associated technological and organizational innovation”* (DACS, 1999).

Curtis (1995) points out that it is difficult to measure cost benefits from process improvements in immature organizations because immature organizations rarely have good cost data (DACS, 1999). Since immature organizations are, by definition, the focus of this study, we felt that we had to devise an instrumentation that would take into account the fact that there would be very little quantitative data available to appraise initial value. Violino (1997) polled 100 IT managers to understand the importance of ROI calculations in IT investments. He found that, “intangible” ROI measures are required to assess a company’s real sources of value. Consequently, our approach is built around an instrument that characterizes the gap between an organization’s current operational state and capability levels targeted in such common models as CMM and ISO 9000-3. The outcome is a single value for a complete set of risk factors that can then be compared against a like value for all of the anticipated returns. McGarry and Jeletic (1993) identified five factors that are required to determine the benefits of process improvement. In turn, we have embodied these within the framework of our assessment instrument:

1. Set goals for what is to be improved,
2. Establish a basic understanding of an organizations current process and product,
3. Investment in change must be made,
4. The effects of the change must be measured to determine if any improvement has been achieved, and

5. Measure the ROI by (a) determining what resources have been expended, b establishing what improvements, both qualitative and quantitative, have been achieved, and (c) determining the difference between the investment made and the benefits obtained

Finally, Capers Jones (1996) has said that process maturity can be assessed based on the degree of planning, sizing, estimating, tracking, measurement infrastructure (development) and reuse activity that is present in the organization and he further ties company size to the range of per employee costs for SPI. He found that organizations typically moved through seven stages on their way to maturity and that depending on the size of the company improvement could take between 26 and 76 calendar months with a return on investment ranging from three-to-one, to thirty-to-one. He says that depending on the stage (the greatest gains coming at level five and above). SPI can result in a 90% reduction in software defects, a 350% productivity gain and a 70% schedule reduction (DACS, 1999). The cost of software process improvement at each of these stages can be uniformly characterized in terms of a common set of factors. 1) *Estimated Cost to reach a given Stage*, 2) *Number of Months to reach a given Stage*, 3) *Estimated Number of Defects*, 4) *Productivity LOC/Day*, 5) *Schedule Length*, 6) *Overall Project Development Costs*, and 7) *Overall Project Maintenance Costs*. The instrument incorporates the first five of these factors as risks (although they could also be treated as benefits). The final two are treated as benefits.

The participants in this appraisal should be senior managers, since Lipke believes that the necessary ingredients for success in SPI are leadership by people at that level (reported in DACS, 1999). In their responses, designated decision-makers are asked to provide their appraisals of such business factors as percentage investment in SPI versus overall investment, degree of current operational performance as characterized by rework, criticality and degree of technical risk assessment. Respondents provide a numeric judgment in response to each question posed. Although the subsequent values are based on estimation the questions are interlocking. Therefore a complete scan is presumed to address every possible contingency of risk versus benefit.

4.0 Evaluating the Risks and Returns of Process Improvement

Given the prior discussion we believe that the instrument and approach that we have developed will successfully evaluate the risks and benefits of software process improvement. Its purpose is to evaluate a range of factors associated with SPI project strengths and weaknesses for risk and return issues. Table one shows how these individual risk and return factors are weighted and scored. The factors in this array are drawn from an OMB study of multiple best practice organizations. Higher scores are given to elements of excessive risk as well as excessive benefit, or those elements that exceed positive aspects of the decision criteria. Additionally, weights have been attached to criteria to reflect their relative importance in the decision process.

TABLE 1: Assessment of Risks and Returns for SPI

Overall Risk Factors: Need for SPI

Factor One: Investment Size	Assigned Value (1 - 10)	x Weight	Criticality
		=	
Estimate the percent of budgeted investment in SPI personnel versus the total budgeted investment in personnel	1 _____ 5 _____ 10 Low % High %	10	<input type="text"/>
Estimate the average hourly rate paid to SPI staff versus average overall hourly rate of pay	1 _____ 5 _____ 10 Low Cost High Cost	5	<input type="text"/>
Estimate percent of source lines of code (SLOC) that will be effected by SPI project in comparison to overall SLOC	1 _____ 5 _____ 10 Large Small	10	<input type="text"/>

Estimate the current Average Defect rate per thousand source lines of code (K)SLOC	1_____5_____10 High Low	10	<input type="text"/>
Estimate the current Software Defect Removal Efficiency Percentage	1_____5_____10 High Low	5	<input type="text"/>
TOTAL FOR THIS ASSURANCE FACTOR			<input type="text"/>

Factor Two: Project Management Process Maturity

Is each project modular (e.g., each project element is Individually planned and resourced)?	1_____5_____10 Modular Non-Modular	3	<input type="text"/>
Is each project schedule based on defined and logically related milestones	1_____5_____10 Consistently Inconsistently	2	<input type="text"/>
Is each project scoped to fit available resources (including staff capability) prior to commitment?	1_____5_____10 Consistently Inconsistently	2	<input type="text"/>
Are project schedules consistently adhered to and the milestone and deadline commitments consistently met?	1_____5_____10 Consistently Inconsistently	6	<input type="text"/>
Are project budget commitments consistently met?	1_____5_____10 Consistently Inconsistently	6	<input type="text"/>
Is software development controlled through use of validated software engineering practices or other disciplined methods?	1_____5_____10 Disciplined Ad-hoc	5	<input type="text"/>
Are inspections consistently carried out for the purpose of identifying problems as early as possible	1_____5_____10 Consistently Inconsistently	3	<input type="text"/>
Are inspections consistently carried out for the purpose of reducing rework	1_____5_____10 Consistently Inconsistently	3	<input type="text"/>
TOTAL FOR THIS ASSURANCE FACTOR			<input type="text"/>

Factor Three: Degree of Technical Risk	Assigned Value (1 - 10)	x Weight	Criticality
		=	

Is the technology base and/or project base primarily geared toward experimental or established technologies?	1_____5_____10 Established Experimental	6	<input type="text"/>
Is the systems architecture and software base technically complex, or routine operational	1_____5_____10 Routine Complex	6	<input type="text"/>

Is there a disciplined management mechanism for integrating new technology and processes into the technology base?	1_____5_____10 Disciplined Ad-hoc	5	<input type="text"/>
Is there a disciplined mechanism for control of change within the technology base (AKA configuration management)?	1_____5_____10 Disciplined Ad-hoc	5	<input type="text"/>
Is there a disciplined mechanism for monitoring, measuring and reporting activity within the technology base (AKA, SQA)?	1_____5_____10 Disciplined Ad-hoc	5	<input type="text"/>
Is the organization's technology base primarily composed of Commercial Off-The-Shelf (COTS) software?	1_____5_____10 COTS Custom Software	3	<input type="text"/>
TOTAL FOR THIS ASSURANCE FACTOR			<input type="text"/>

OVERALL ASSURANCE SCORE	<input type="text"/>
-------------------------	----------------------

Overall Return Factors

Business Impact	Assigned Value (1 - 10)	x Weight =	Criticality
Is the investment in SPI aimed at improving the performance of a specific area of the organization?	1_____5_____10 Specific Overall	10	<input type="text"/>
Can the benefits of SPI be expressed in outcome oriented terms?	1_____5_____10 No Yes	5	<input type="text"/>

Customer Needs -	Assigned Value (1 - 10)	x Weight =	Criticality
Can the investment in SPI be referenced to identifiable internal and/or external customer needs or demands	1_____5_____10 Unrelated Related	10	<input type="text"/>
Have internal and/or external customers reported problems with quality and/or timeliness of delivered software product	1_____5_____10 No Yes	10	<input type="text"/>

Return on Investment	Assigned Value (1 - 10)	x Weight =	Criticality
Are cost-benefit analyses performed before committing to each Project	1_____5_____10 Yes No	3	<input type="text"/>
Are technical needs, or considerations the primary driver for commitment decisions	1_____5_____10 No Yes	2	<input type="text"/>

Are project commitment decisions reviewed and/or authorized by managers above the technical level	1_____5_____10 Yes No	2	<input type="text"/>
Does the organization primarily obtain its software from acquisition rather than development	1_____5_____10 Yes No	3	<input type="text"/>
Are cost-benefit results reliable and technically sound	1_____5_____10 Solid Risky	5	<input type="text"/>
Can the investment in SPI be shown to result in a reduction in costs?	1_____5_____10 Unclear Demonstrated	10	<input type="text"/>

Organizational Impact	Assigned Value (1 - 10)	x Weight	=	Criticality
------------------------------	-------------------------	----------	---	-------------

Does the SPI Project affect a large part of the organization (<i>i.e., a large number of users, work processes, and systems</i>)?	1_____5_____10 Low Impact High Impact	15	<input type="text"/>
---	---	-----------	----------------------

Improvement Context	Assigned Value (1 - 10)	x Weight	=	Criticality
----------------------------	-------------------------	----------	---	-------------

Is the SPI effort intended to support, or enhance an existing operation or is it intended to improve future capability	1_____5_____10 Tactical Strategic	5	<input type="text"/>
Is the SPI effort necessary to meet the requirements of a contract or other externally mandated requirement	1_____5_____10 Internal External	10	<input type="text"/>
Is the SPI project required to maintain the organization's critical functioning	1_____5_____10 Not Critical Critical	5	<input type="text"/>
Is the SPI project expected to produce a high level of improvement	1_____5_____10 Low Level High Level	5	<input type="text"/>

OVERALL RETURN SCORE	<input type="text"/>
-----------------------------	----------------------

RETURN ON INVESTMENT <i>(the RETURN SCORE minus the ASSURANCE SCORE)</i>	<input type="text"/>
--	----------------------

The respondent provides a one-to-ten numeric response for each question. When that is arrived at this value is multiplied by the weight assigned to it and the calculated total is placed in the boxes provided opposite the question. Once each section is completed, the values that have been calculated and placed in each individual box are summed and entered in the box next to "Total for this Factor". These factor scores are summed to obtain a total

section score. Once a score is obtained for both the Risk and the Return sections the Overall **Risk Factor** score is subtracted from the Overall **Return Factor** score to arrive at a ROI value estimate. Then:

- If that number is positive there is likelihood that you will get a positive return on investment from a Software Process Improvement Project.
- If that number is negative there is likelihood that the investment in an SPI program will not generate a worthwhile return.
- If the value obtained is greater than 100 (e.g., a 10% difference in either direction) than it is strongly recommended that the result be considered indicative for the purpose of decision-making.

As noted earlier, although the assigned values are based on estimation the questions are interlocking. Therefore a complete scan of an operation is presumed to address every possible contingency of risk versus benefit.

5.0 Instrument Validation

In order to assess this instrument’s reliability, we asked representatives of three different organizations (one car company, one first tier supplier and a defense contractor) to fill it out. We were looking to confirm two factors: inter-rater reliability and the predictive power of the instrument. The first factor was investigated by way of a correlation of the responses of decision-makers from the same unit in three different corporations. Essentially we asked IT employees at similar levels and in similar places to rate their company based on the instrument. This rating was collected blind (e.g., the raters did not communicate with each other during the rating process) and simultaneously. Table Two presents the results:

TABLE 2: Inter-rater Responses

Supplier One	Rater 1	Rater 2	Rater 3	Rater 4	Rater 5	Rater 6
<i>Investment Size</i>	178	211	180	175	270	166
<i>Process Maturity</i>	210	235	244	239	250	230
<i>Technical Risk</i>	170	223	168	190	173	160
<i>Business Impact</i>	150	150	87	110	118	90
<i>Customer Needs</i>	186	171	147	212	122	99
<i>Cost-Benefit</i>	152	126	110	144	127	87
<i>Organizational Impact</i>	100	95	139	104	167	77
<i>Improvement Context</i>	205	199	187	168	212	164
Correlation		0.852		0.776		0.825

Defense Two	Rater 1	Rater 2	Rater 1	Rater 2	Rater 1	Rater 2
<i>Investment Size</i>	200	222	111	166	220	265
<i>Process Maturity</i>	250	201	187	202	199	254
<i>Technical Risk</i>	268	220	118	170	97	222
<i>Business Impact</i>	150	150	87	117	84	114
<i>Customer Needs</i>	200	170	134	212	120	155
<i>Cost-Benefit</i>	171	144	90	126	77	125
<i>Organizational Impact</i>	115	87	77	122	54	99
<i>Improvement Context</i>	187	196	101	187	185	132
Correlation		0.853		0.786		0.721

Car Company Three	Rater 1	Rater 2	Rater 1	Rater 2	Rater 1	Rater 2
<i>Investment Size</i>	109	210	163	184	263	228
<i>Process Maturity</i>	158	235	145	222	120	164
<i>Technical Risk</i>	172	240	157	170	137	190
<i>Business Impact</i>	110	161	130	110	123	97
<i>Customer Needs</i>	188	173	145	200	221	183
<i>Cost-Benefit</i>	122	142	113	130	227	187
<i>Organizational Impact</i>	90	115	89	100	45	66
<i>Improvement Context</i>	126	127	155	165	137	204
Correlation		0.576		0.742		0.776

As can be seen with a couple of exceptions there is a surprisingly high degree of relationship between raters. Because these values were so high we actually followed up with each of our raters to double check the application of our protocol. All of the raters felt that the results correctly reflected their perception of their unit's situation and none of the raters was aware of how the other person in their unit had scored the instrument (although several expressed no surprise that the results were so similar).

The second, and perhaps more interesting aspect of this study was the attempt to validate the predictive power of the instrument. The ratings listed above were compared to the known level of process maturity of the corporations for which they were prepared. The first corporation is ISO 9000 certified. The second corporation has been consistently assessed at CMM level three. While the IS&S area in the third company is a classic CMM level one chaos operation. Looking at the scores obtained during our validation study we found that there was considerable variance in the assessed need for formal process improvement (based on our scoring system) among these three. Table Three outlines this:

TABLE 3: Comparison of Ratings

	Average Risk	Average Return	Average Difference
Supplier One	612	701	89
Defense Two	595	661	66
Car Company Three	545	697	152

As can be seen, the company that is arguably in the position to benefit the most from a formal SPI process is also the one that has the highest rating for return on that investment (15.2%). Whereas, the score for the company that we call Defense Two (which is already at level three CMM) actually seems to indicate that additional expenditure for formal SPI in this company would probably not produce a sufficient enhancement of the current operation to justify the investment (6.6%).

We were particularly interested in the score for Supplier One. It is one of the small set of IT operations in our area that is fully ISO 9001 certified and the relatively low differential score appears to reflect this. Furthermore given the general belief that there is a reasonable degree of correspondence between an ISO 9000 operation and CMM level two, this also tended to substantiate our notion that this instrument is correctly responding to variations in process maturity.


The important fact from the perspective of this article however, is the evidence that the questionnaire appears to be internally consistent as well as predictive. This first study was not rigorous enough to be claimed as scientific, so we are not peddling these results as conclusive. However the consistently reasonable correlation

between raters as well as the apparent ability to differentiate companies based on their known levels of process maturity is encouraging to the authors.

6.0 Summary and Conclusions

This article has presented a simple mechanism and approach that will allow decision-makers to evaluate the return on investment of SPI prior to launching such an effort. Why is this potentially useful? First and foremost it is worthwhile because the attitudes in the IT community toward software process improvement generally approach religious zeal. That is, people either believe in it, or they don't. But there has never been a lot of "thinking through" of how, or whether, an organization arrived at their given doctrine. This is dangerous for a lot of reasons, the most obvious and practical one merely being the question of whether the organization will then spend its precious resources wisely.

That question should keep decision-makers up at nights because SPI costs money and in some cases that expense can be significant. Obviously, it will be easy to tell ten years up the road whether the right decision was made. But a CEO, or CIO contemplating laying out six, or seven figures for the additional personnel and resources to conduct SPI is not in a position to make that call and the wise ones will not be led into it by blind faith. Which implies the need for some sort of reliable crystal ball.

Our bias in the beginning was that any expenditure in formal SPI is money well spent. However, our own short validation study encouraged us to temper that enthusiasm with a little realism. Oddly enough this new caution was also substantiated by some early studies done at the Software Engineering Institute (reported in SEI, 1989). These tended to indicate that the gap between the most effective IT organizations and the ones that were the least capable was remarkably wide. The problem was sorting out which was which and putting their relative position into some sort of perspective that a non-technical decision-maker could understand. We believe our instrument serves that purpose. It appears in the first cut to successfully identify organizations in need of initial SPI and it also appears to identify those where such an effort is not worth the cost. We will continue our validation studies, but if this ability to discriminate holds up we believe that this could be a valuable tool for organizations trying to invest wisely in the IT marketplace. 

References

1. Broadman JG, Johnson DL, *Return on Investment from Software Process Improvement Measured by US Industry*, Software Process - Improvement and Practice, July 1995
2. Brynjolfson Erik, "The Productivity Paradox of Information Technology", Communications of the ACM, Vol. 36, No. 12, pp. 67-77, December 1992
3. Construx Software Builders, web site @ www.construx.com, 1998
4. Curtis W., *Building a Cost-Benefit Case for SPI*, 7th SEPG Conference, Boston, 1995
5. *Evaluating Information Technology Investments*, Office of Management and Budget, at www.itmweb.com, 1999
6. Hersleb, J, Zubrow D, Siegel J, Rozum J, Carlton A, *Software Process Improvement: State of the Payoff*, American Programmer, v7 no.9, September, 1997
7. Humphrey Watts S., *Managing the Software Process*, Addison-Wesley: Reading, MA, 1994
8. Jones Capers, *The Pragmatics of Software Process Improvements*, Software Engineering Technical Council Newsletter, No. 5, Winter 1996
9. KPMG Technology and Services Group, web site at www.kpmg.ca 1998
10. Laker Consulting, web site at www.laker.com.au., Sydney, 1998
11. McGarry F. and K. Jeletic, *Process Improvement as an Investment: Measuring its Worth*, NASA Goddard Space Flight Center, SEL-93-003, 1993
12. McGibbon, Thomas, *A Business Case for Software Process Improvement Revised*, DoD Data Analysis Center for Software (DACS), 1999
13. O'Brien M, *Software Production Management*, NCC Blackwell Ltd.: Oxford, U.K., 1992
14. Paulk M., B. Curtis, M. Chrissis, C.Weber, "Capability Maturity Model, Version 1.1," Technical Report,

15. Software Engineering Institute, Carnegie-Mellon University, 1993
Software Engineering Institute, web site at www.sei.cmu.edu. 1998
16. Strassman, P.A., *The Business Value of Computers*, The Information Economics Press, New Canaan, Connecticut, 1990
17. Stephen S. Roach, "Services Under Siege-The Restructuring Imperative." Harvard Business Review, pp. 82-92, Sept.-Oct., 1991
18. Violino R, *Measuring Value: Return on Investment*, Information Week, No. 637, (June 30, 1997) pp. 36-44
19. Yamamura G, Wigle GB, *SEI CMM Level Five: For the Right Reasons*, Crosstalk, Volume 10#8, August 1977

Notes

Notes