Maturity and its Impact on

New Product Development Project Performance

Kevin Dooley

Departments of Management and Industrial Engineering

Arizona State University

Anand Subra

IBM Corporation

John Anderson

Department of Operations Management

University of Minnesota

Cite: Dooley, K., Subra, A., and J. Anderson (2001), "Maturity and its Impact on New Product Development Project Performance," *Research in Engineering Design*, 13: 23-29.

corresponding author:
Prof. Kevin Dooley
Arizona State University
PO Box 874006
Tempe, AZ 85287-4006

running title: Maturity and New Product Development

Maturity and its Impact on New Product Development Project Performance

Abstract

The Software Engineering Institute's Capability Maturity Model (CMM) has popularized the concept of process maturity. Software organizations use the CMM to assess their current capabilities and plan for further improvements. The purpose of this paper is to generalize the concept of maturity beyond the software engineering domain and determine the impact of maturity on project performance in new product development. In this paper we define the construct of maturity as the degree to which a process is defined, managed, measured, and continuously improved. We employ survey methodology to operationalize the construct, and a sample of 39 new product development programs is used to test the proposition that increased maturity leads to better project results, as measured by project cost and timeliness. We also test a number of environmental factors as moderators of this relationship. Results confirm a positive relationship between maturity and project success.

Problem Statement

The quality and timeliness of the new product development (NPD) process has been hailed as a critical strategic factor for the success of organizations (NRC, 1991). High-quality NPD processes have been shown (e.g. Cooper, 1993) to be associated with numerous beneficial organizational outcomes. Traditionally, improvement in NPD has been sought through implementation of *best practices.* In general, a practice is a tactic or method chosen to perform a particular task, and/or to meet a particular objective. For example, selection of project team members would be a practice that would help define the ability of the organization to be capable of rapid response. A best practice is a tactic or method that has been shown through real-life implementation to be successful; a cross-functional development team would be an example of a best practice (Olson, Walker, & Ruekert, 1995). Various research studies (via description) and design gurus (via prescription) have identified a large number of such best practices that could beneficially be implemented in the organization's NPD processes (e.g. Cooper, 1993; Dixon & Duffey, 1990; Paulk et al., 1993).

Best practices--or *what* is done--only define one component of success. Equally important are the issues of *maturity* (how well the system does what it does) and *diffusion* (how widely and how often the organization performs the best practice). It is hypothesized that widely diffused best practices in a mature NPD process will lead to greater organizational effectiveness and more successful products.

The concept of maturity is an integral component of the Software Engineering Institute's (SEI) process Capability Maturity Model (CMM) (Paulk et al., 1993). The CMM is used by the U.S. Department of Defense to evaluate the capabilities of software suppliers. The maturity of an organization's software development process is assessed to be at one of the following five levels: *initial, repeatable, defined, managed and optimizing*. In the *initial* level the development process is ad hoc; there is no synergy between development projects, and work succeeds simply because of the effort of individuals. As the system gains some maturity, the development process becomes *repeatable*. Efforts and actions can be replicated with some consistency across multiple projects. As the process is *defined* it takes on an even greater degree of consistency, repeatability, and in some sense, predictability. As the process becomes *managed,* various performance measures are collected and used to track process performance. Project data is collected and analyzed over multiple projects, and some corrective action takes place. At an *optimizing* level, interim measures of project (process) performance are monitored for potential immediate corrective actions. A multi-project view becomes an integral part of a proactive, as opposed to reactive, quality improvement strategy.

This paper concerns itself with the theoretical development of the construct of maturity. First, the CMM's formulation of the maturity construct will be described; relationships between software engineering maturity and organizational performance will be summarized. Next, a more general definition of maturity will be proposed, namely, that maturity is the degree to which a process is defined, managed, measured, and continuously improved. We employ survey methodology to operationalize the construct, and a sample of 39 new product development programs is used to test the proposition that increased maturity leads to better project results, as measured by project cost and timeliness. We also test a number of environmental factors as moderators of this relationship.

The Maturity Construct in Practice

The Software Engineering Institute's (SEI) process Capability Maturity Model (CMM) (Paulk et al., 1993) is used by the U.S. Department of Defense to evaluate the capabilities of software suppliers. It loosely has roots to Crosby's Quality Management Maturity Grid (Crosby, 1979). The model assesses a software development organization to one of five levels of process maturity (initial, repeatable, defined, managed and optimizing), and thus represents a way to gauge/predict performance on an ordinal scale. However, it has been used almost exclusively in the software engineering environment. The capability maturity model is descriptive and normative, and defines the key practices that characterize and differentiate each successive level of process maturity. Thus, it provides a method for assessing firms' capabilities to produce quality software. The CMM defines maturity as "the extent to which a specific process is explicitly defined, managed, measured, controlled, and effective" (Paulk et al., 1993, sec. 1.2).

The CMM's maturity levels are defined as:

• Level 1 - *initial* - The process is characterized as ad hoc, and occasionally even chaotic. Few processes are defined, and success depends on individual effort.

• Level 2 - *repeatable* - Basic project management processes are established to track cost, schedule, and functionality. The necessary process discipline is in place to repeat earlier successes on similar projects.

• Level 3 - *defined* - The process for both management and engineering activities is documented, standardized and integrated into a standardized process for the organization. All projects use an approved, tailored version of the organization's standard process. Activities are well integrated.

• Level 4 - *managed* - Detailed measures of the process and product quality are collected. Both the process and products are quantitatively understood and controlled.

• Level 5 - *optimizing* - Continuous process improvement is enabled by quantitative feedback from the process and from piloting innovative ideas and technologies.

Key practices that differentiate each succeeding maturity level are articulated. Levels are inclusive, so Level 2 practices must be essentially in place before Level 3 maturity can be achieved.

Another way of viewing the maturity levels is that each represents a different degree of visibility into the process. At Level 1, the process is a black box. At Level 2, the process is visible at certain transitional periods denoted by stage gates or project milestones. At Level 3, the internal mechanisms at each stage become visible. At Level 4, extensive performance measurement takes place, and improvements focus on the steps involved in each development stage. At Level 5, the structure of the process is continuously improved (Paulk et al., 1993, sec. 2.3).

CMM is made operational in the following manner. The maturity level is indicative of process capability, and contains key process areas. For example, Level 2 (repeatable) is indicative of a disciplined process, and contains key process areas such as software project planning, requirements management, and software project tracking and oversight. A key process area achieves goals, and is organized by common features. Common features include (a) commitment to perform, (b) ability to perform, (c) activities performed, (d) measurement and analysis, and (e) verifying implementation. Each common feature contains key practices that describe activities to be undertaken. For example, a key practice addressing implementation of software project planning is "estimates for the size of the software work products are derived according to a documented procedure" (Paulk et al., 1993, sec. 3.3). It is at the level of key practices that the organization is typically audited. Audit results are compiled by an external team and shared with organizational members. Such audit results would typically be incorporated into the organization's next strategic plan.

From 1987 to 1994, a total of 379 organizations have been formally assessed by SEI; upon initial assessment, the distribution of maturity was 73% initial, 16% repeatable, 10% defined, 0.6% managed, and 0.3% optimizing (Hayes & Zubrow, 1995). It has been found that project management--project planning, tracking, and oversight specifically--is the key differentiator between level one and level two performers (Hersleb et al., 1994). Organizations took on average 25 months to move from one level of maturity to the next (Hayes & Zubrow, 1995).

The median value of return on investment in software NPD improvement has been 500 percent, from an annual investment of $245,000 (Hayes & Zurbow, 1995). Previous studies found CMM maturity level to be positively correlated with ability to meet schedule and financial targets, product quality, staff productivity, staff morale, and customer satisfaction (Goldenson and Hersleb, 1995). An organizational study of 176 software engineering groups showed that maturity level was positively correlated to reduced employee turnover and absenteeism, reduced software defects after release, decreased cost of poor quality, improved competitive position, decreased cycle time, decreased development cost, and decreased schedule and cost overruns. The study also found that while about 20 months were required to move from Level 1 to Level 2, fewer months were required for each subsequent step (Williams, 1997).

It should also be noted that the CMM framework has been adapted for application to systems engineering (Kuhn et al., 1996) and organizational development (Curtis, Hefley, &

Miller, 1995). While specific details differ between these frameworks and the CMM, the general definition of the levels of maturity remain the same.

A Theoretical Model of Maturity and Project Performance

This paper seeks to contribute to existing knowledge and conceptualizations involving *maturity* in two ways. First, while there are attempts to generalize the concept of maturity beyond the domain of software engineering (Kuhn et al., 1996), the concept itself has not been defined in a manner that would enable it to be researched via large-scale empirical studies. Second, no studies outside of software engineering have linked maturity and project performance. In this paper we develop survey items to measure maturity and project performance, and then correlate the two empirically.

The CMM formulation of the maturity construct appears to have good predictive validity within the domain of software engineering, so it is reasonable to use it as a starting point for a more general formulation of the construct. In fact, we only suggest two alterations to CMM's nominal definition of maturity: "the extent to which a specific process is explicitly defined, managed, measured, controlled, and effective" (Paulk et al., 1993, sec. 1.2). First, the CMM definition implies that maturity is both a set of process activities (defined, managed, measured, and controlled) as well as process outcomes (effective). A better conceptualization should separate activities from outcomes. Second, the term *controlled* only implies elimination of special causes of variation, and not necessarily improvement of the common cause system. In defining maturity we wish to capture both the quality control and quality improvement aspects of the quality system. We thus will substitute the term *continuously improved* for the term *controlled*. Therefore we nominally define maturity as *the extent to which a process is explicitly defined, managed, measured, and continuously improved.*

Note that this nominal definition implies simultaneity to the activities of definition, management, measurement, and continuous improvement. This is in contrast to the CMM levels which imply a sequential relationship between these activities. It is our formulation however that these elements typically co-exist and there is no reason to define maturity more strictly as a process model. Dekleva and Drehmer (1997) investigate the time at which certain software engineering practices are implemented in firms and show that while there is some sequence-dependency, as predicted by the CMM levels, there is also a large amount of overlap, i.e. even when an organization has practices primarily indicative of a given level of maturity, practices indicative of a higher level of maturity may simultaneously be present.

Project performance is often measured in terms of schedule, cost, and other organization objectives (Griffin, 1997). For example, the U.S. Department of Defense guide to integrated product and process development (DoD, 1996) states the purpose of the development process is "to meet cost and performance objectives" (p. 2-7); their emphasis on "better, faster, and cheaper" (p. 2-9) echoes these metrics. We nominally define project performance as *the degree to which project schedule, cost, and other objectives are met.*

The linkage between maturity and project performance can be reasoned as follows. Consider first the normative case. An organization would first define their development process in terms of required and optional steps, necessary decision points, patterns of information flow and coordination, and a resource allocation scheme. Once a standard process is in place, meaningful data could be collected. Data from such a standardized process would be relatively easy to interpret because the majority of its variation should reflect the common cause system, e.g. because the process is standardized, only common causes of variation should be present within the data. Common causes are those sources of variation that are indicative of the routine process (Deming, 1986). For example, variation in task completion times would be due to sources of variation such as personnel, procedures (at a micro level), and infrastructure. As special causes occurred, the organization could determine it from the data and take subsequent corrective action. Special causes indicate the breakdown of the routine process (Deming, 1986). Having used both an approach of process standardization as well as statistical process control to reduce the process variation to common causes only, corrective actions could be taken in terms of changing the standard process. In doing so, the organization would set goals to be associated with each of the performance measures. The comparison of metrics to goals would indicate where the organization needed to focus, and would help them learn about the intricacies of the process.

This learning process should facilitate the reduction of variation in the process, as well as enhance participant understanding of the capabilities of the development process. Reducing variation in the process should positively impact cost and schedule, since many cost and schedule overruns are due to special causes or excessive variation in the common cause system. Enhancing participant knowledge of process capability will lead participants to set more realistic cost and schedule objectives, thus increasing the likelihood of project success.

If any of the components (defined, managed, measured, and continuously improved) is missing, project outcomes may suffer. For example, if data is collected from an non-standardized process, it will be of little use in analysis. The sources of variation present in the data will include differences between projects in terms of how they were executed (at a macro rather than micro level). If a special cause does occur, or if a troublesome source of variation is present, it will unlikely be seen because there will be no way to differentiate it from the variation due to different executions. If learning (controlling) is attempted without metrics in place, then learning will be based on intuition. While a great deal can be gained from the expert insight of people working within and on the system, data is necessary in order to counter-balance the bounded rationality of human decision makers.

Other studies have established some credence to there being a relationship between the elements of maturity and project performance. A number of studies have indicated the benefits of structured processes (Booz, Allen, and Hamilton, 1982; Mercer Management Consulting, 1994; PRTM, 1995). Griffin (1997) found a structured, formal development process helps reduce development cycle time, and this effect is accentuated when the project involves the development of a relatively complex product.

Eisenhardt and Tabrizi (1995) found in a study of the computer industry that firms using an "experiential strategy of multiple design iterations, extensive testing, frequent project milestones, a powerful project leader, and a multifunctional team" (p. 84) accelerated development, thus pointing to the importance of measurement and continuous improvement via experiential learning cycles. The consulting firm PRTM (1995) found that higher performing firms measured both product and project performance. Griffin (1997) found that "best practice firms are more likely to measure NPD performance" (p. 431). Van de Ven and Polley (1992) found linkages between goals, measurement, and learning. In a longitudinal study of learning in a development project, they found learning to be absent during the initial stages of the innovation process, when goals were ambiguous and divergent and explicit measurement of performance did not exist; in the later stages of development when goals and performance measurement were in place, learning was observed. Finally, the DoD guide (1996) for integrated product and process development specifically prescribes that "…continuous, measurable improvement should be an integral part of (NPD) implementation. Defining and using process-focused metrics allows for early feedback and continuous monitoring and management of (development) activities and program maturation" (p. 2-7).

Thus we make the following proposition:

Proposition 1: Maturity in the new product development process will be correlated positively with project outcomes.

It is worthwhile to consider how this relationship might not be true. First, it is well known that project and product outcomes are only loosely associated. In responding to perceptions about process outcomes, participants may attribute poor product outcomes with poor process outcomes. Thus in situations were product outcomes were poor, respondents may respond that process outcomes were poor, even though maturity indicators are responded to positively. Second, firms may structure and develop a quality system around a NPD process that is inherently of the *wrong type*. For example, a firm may use a structured, stage-gate approach where a more experiential, prototype-based process may be applicable. Thus, structure and improvement may have limited impact on actual process outcomes. Third, since process outcomes are measured relative to expectations, they may be deemed as always being negative if expectations concerning budget and timeline are unreasonable.

The effect of maturity on project performance may be contingent on certain environmental factors. For example, smaller companies may have projects that are also smaller in scope. These projects may be able to be successfully managed in an ad hoc manner, whereas the more complex projects typically found in a larger organization may require some level of structure in order to succeed. Eisenhardt and Tabrizi's study (1995) hinted that maturity (as defined by a structured process and a "rational" improvement approach) might be less important in volatile industries, where instead a process of rapid prototyping and experiential learning is most fit. Finally, new product development programs can be distinguished by their customer--some programs produce products primarily for an end-consumer market, while others produce product primarily for an industrial market. Because NPD process maturity is an internal issue, and one that is likely invisible to the end customer, the relationship between maturity and project outcomes should be the same regardless of customer type.

Proposition 1a: Company size positively moderates the positive relationship between maturity and project outcomes.

Proposition 1b: Market volatility negatively moderates the positive relationship between maturity and project outcomes.

Proposition 1c: The positive relationship between maturity and project outcomes will be identical for industrial and consumer companies.

The next section explores the constructs are made operational, and the experimental design used to test the proposition.

Measuring the Maturity Construct

We developed items for measuring the *maturity* construct directly from our nominal definition. Two items (each) were chosen to be indicative of activities concerning process definition, management, measurement, and continuous improvement. We are not hypothesizing a four-dimensional construct; rather we believe these represent different facets of a mature process and therefore will psychometrically form an eight items, one-dimensional scale. The scale items for the construct *project outcomes* were similarly developed, directly from the nominal definition. Moderator variables were measured with a single item question. Content validity of the constructs was conducted via a check of face validity; five engineering managers were asked to assess the items and determine whether or not the list was complete. The existing items represent the outcome of that feedback.

The items were worded in such a manner that measurement of maturity takes place at the level of a development program. A NPD program is a collection of projects that share common features and processes. In any large organization there may exist several programs, typically differentiated by product sector. It is necessary to measure maturity at a program rather than project level, since the dimensions only exist in the context of multiple projects, i.e. there is no sense in discussing the maturity of a single development project. A Likert scale was used to measure respondent's perceptions; roughly this meant that a response of “0” meant that the particular practice in question was not used anywhere in the organization, and a response of “5” meant that the practice was used on all development projects (within a time frame of the last two years).

A population of organizations was constructed using personal contacts from the researchers, with equal representation from the commercial and industrial sectors. Requests for participation were also posted in various electronic forums. A total of 39 organizational respondents were generated, from an original mailing of 250, for a response rate of 16%. The firms represented a wide variety of SIC codes, but all could be considered in the general product area of electro-mechanical devices. Six of the 39 firms had revenues in the range of $1-10 million; 10 had revenues in the range of $10-100 million; and 23 had revenues over $100 million. Thirty of the 39 considered themselves to be in the top one-third of the market, in terms of market share. Two-thirds of the firms had one at least one quality award (state, national, supplier) for their quality performance. Two-thirds of the firms were certified in ISO 9000, and an equal amount was self-described to be in markets that could be described as "volatile and growing".

No formal analysis of non-respondents was performed; informal analysis of non-respondent demographics demonstrated no difference with the respondents. Additional evidence points towards non-respondent bias not being a problem. The average response for project outcomes was 2.94 on a 1-5 scale, indicating that if there was any non-respondent bias, it didn't have to do with performance. No explicit measures of criterion-related validity were considered.

Each survey was completed (typically) by a single individual, most often the VP or Director of Engineering or Marketing. Single respondent problems exist (DeVellis, 1991). In order to formally test whether the singular responses would have an influence on subsequent data analysis, two organizations were asked to have five individuals complete the survey. Analysis of variance was then performed, and it showed that the variation of (average) scores between organizations was significantly greater than the variation between respondents in a single organization. Thus, we assume that the responses from the individual are reasonable assessments of what was actually happening.

Results

Standard survey scale analysis was performed (DeVellis, 1991). First, for each of the two scales, Cronbach's alpha was calculated as a measure of reliability, indicative of construct validity. Next, factor analysis (with a varimax rotation) was used to confirm the unidimensionality of the constructs. A summary of the final scales is shown in Table 1. One can see that the two constructs are identified separately as two factors, indicating divergent validity of the constructs; and as they are unidimensional and have high Cronbach alphas, construct validity is considered adequate.

Statistical regression was used to test the propositions posed earlier (see Table 2). None of the interaction terms, representing the propositions regarding moderating variables, are statistically significant. Maturity does regress successfully on project success. The model is statistically significant ($p<0.00029$) as is the parameter corresponding to maturity ($p=0.000287$); the normalized beta-value is 0.51. The model captures a significant portion of the observed variation in the data; r^2 corrected is 0.32. Analysis of residuals shows no transformation of data is required, and no outliers are present.

--insert Table 1—

--insert Table 2—

Discussion

Our results indicate that the presence of a defined, managed, measured, and continuously improved new product development process is positively correlated with project success, as defined by meeting cost, schedule, and organizational objectives. This correlation does not appear to be contingent on organizational size or the volatility of the market the firm is in, and can be generalized to both industrial and consumer sectors. Our sample population represented a diversity of industries across several SIC codes, but most could be described as making electro-mechanical products, thus our results can be generalized specifically to those segments of industry.

Our results lend credibility to prescriptions calling for quality systems methodology and philosophy to be applied to the new product development process (Clausing, 1994). It also supports the notion that *maturity* is a valid and important concept outside of the domain of software engineering, thus signaling the potential to develop capability maturity models for other domains. For example, Baumann et al. (1999) have developed a network maturity model, based on the assumption that the maturity of the development process and operational processes of telecommunications and computer networks will have a positive impact on the organization.

Any prescriptions for maturity depend on an assumption of causality between maturity and project outcomes; causality cannot be determined in this study, however, only correlation. A positive correlation may have been observed because cause and effect are opposite proposed: successful projects generate both organizational learning and organizational slack, enabling workers to have the time and ability to implement mature processes. The positive correlation may also have been observed because participants have self-selected memories (March et al., 1994). In responding to the survey, their memory of outcomes (project outcomes) might be much better than their memory of details (maturity), thus memories of positive project outcomes may lead to memories, perhaps inaccurate, of high maturity, and visa versa.

Examination of the data case by case, as well as some anecdotal data collected by the authors in the course of this investigation indicate that many organizations collected data from unstructured, undefined NPD processes. The end result is likely to be frustrating. Such data will contain so many sources of variation that it will likely be impossible to discern much of value, except very broad-scale trends over longer periods of time.

This case-by-case data and anecdotal evidence also point to the observation that many organizations may not be learning much about how to improve their development processes, even though they are collecting process data. There are two possible explanations. First, it may be that the data is coming out to be somewhat meaningless because it is being collected from an unstructured process (as discussed above). Secondly, it may be that much performance measurement of NPD is likely to be inadequate because of the nature of complexity within NPD. Data collection and analysis only leads to learning if clear cause and effect relationships can be determined. The nature of NPD may be such that clear cause and effect relationships, at least in the traditional sense, are very difficult to determine or uncover.

There is some evidence that organizations are growing weary of the benefits of measuring NPD performance. Griffin (1993) reports that:

- Firms that measure NPD success and failure are less likely to want to develop firm-level performance metrics than those that currently do not collect data.
- Firms that measure success and failure cite a "lack of accountability for results" as a reason for why they don't measure what they want.
- Technology-based firms are more likely to not measure what they want because "they do not understand the development process".

Each of these findings is consistent with our interpretation. First, there is more optimism for measurement (and subsequent learning) from organizations that have not yet established measurement systems; those that have established measurement systems do not share the same degree of optimism. Second, lack of accountability for results may have to do with the fact that metrics point towards such a complex web of cause and effect that no determinable cause can be found. Last, technology-driven (as opposed to market-driven) organizations are more likely to have rather complex development processes; the complexity of such a process, especially if it has not been standardized, is likely to be beyond the comprehension of the individuals working with it. One is tempted into question the validity of organizational actions whereby significant efforts are being placed into the collection of metrics which are not leading to learning, either because of they complexity of the NPD process, and/or because organizations are not taking the first basic step of variance reduction by standardizing their processes.

Let us suppose for the moment that the NPD process really is a very complex web of cause and effect relationships. Why is it still not feasible to measure its behavior and learn from such measurement? Consider an organization that decides it will standardize part of its process by implementing design rules. These design rules establish the protocol for interaction between different design groups (for example, between mechanical and electrical design). Such design rules help implement the concept of modular design, where components of a product can be designed independently because their interactions are kept to a minimum, and whatever interactions are present are managed by design rules. The establishment of these design rules may have no noticeable effect for quite some time. With a traditional linear cause and effect lens, managers overseeing this investment would observe that the implementation of design rules had no effect on outcomes such as market share, etc. The lack of influence may have been contingent on the organization effectively taking advantage of the modular design approach however. Perhaps as the organization designs a new product platform that takes advantage of modular design, benefits will be gained.

If NPD processes are realistically represented by complex cause and effect relationships, is it possible to gain anything from measurement? Yes, it still is possible to learn from measurement; a few suggestions may be pertinent (to both researchers and practitioners):

- Qualitative (as opposed to quantitative) measurement of complex systems can often lead to significant learning. Kleiner and Roth (1997) have developed a process, for example, whereby rich, qualitative cases can be written and shared with organizational members for the purpose of learning. These cases contain not only a summary of the "history" of activity, but also the reflective comments of participants and observers.

- Measurements should be separated into similar contexts. For example, data taken from development projects with similar characteristics (in terms of complexity of design, amount of interaction with customer and supply chain, etc.) are more likely to yield results that can be interpreted with some meaning, leading to significant learning.
- The actors involved in the process, not only by managers overseeing the process, should interpret measurements. Learning is most effective if the interpreters have immersed themselves in the activities they are studying.

Summary

In summary, this paper has put forth the notion than the maturity construct--used extensively by the software engineering field--can be generalized outside of that domain. Maturity consists of having a defined, managed, measured, and continuously improved process--these are the essential elements of any quality system. A survey instrument was developed to measure the relationship between maturity and project performance, and it was found that higher levels of maturity were associated with projects that met cost, schedule, and other organizational objectives. This effect is not contingent on organizational size or the volatility of the market the firm is in, and is can be generalized across both industrial and consumer sectors.

Acknowledgements

This research was supported by the National Science Foundation (NSF 9529921), Honeywell, and 3M Corporation. Thanks are due also to the anonymous reviewers who have made valuable suggestions for improvement.

Bibliography

Baumann W, Bhattacharya S, Capone J, Dooley K, Fritsch J, Palangala S (1999) The network maturity model for internet development. IEEE Computer October: 117-118

Booz, Allen, and Hamilton (1982) New products management for the 1980's. Booz, Allen, and Hamilton, New York.

Clausing, D (1994) Total quality development. ASME Press, NY

Cooper, RG (1993) Winning at new products, 2nd ed., : Addison Wesley, Reading, Massachusetts

Crosby, P (1977) Quality is free, McGraw-Hill, NY

Curtis, B, Hefley, W, and Miller, S (1995) Overview of the people capability maturity model, CMU/SEI 95-MM-01, Carnigie-Mellon Software Engineering Institute

Dekleva, S, and Drehmer D (1997) Measuring software engineering evolution: A Rasch calibration. Information Systems Research 8(1): 95-105

Deming, WE (1986) Out of the Crisis. MIT Press, Cambridge, MA

Department of Defense (U.S.) (1996) DoD guide to integrated product and process development (version 1.0)

DeVellis, R (1991) Scale development:Theory and applications. Sage Publications, London

Dixon, JR and Duffey, M (1990) The neglect of engineering design. California Management Review Winter: 9-23

Eisenhardt, K, and Tabrizi, B (1995) Accelerating adaptive processes: Product innovation in the global computer industry. Administrative Science Quarterly 40: 84-110

Goldenson, D, and Herbsleb, H (1995) After the appraisal: a systematic survey of process improvement, its benefits, and factor for success. SEI/CMM-95-TR-009 Carnegie-Mellon, Software Engineering Institute

Griffin, A (1993) Metrics for measuring product development cycle time. Journal of Prod. Innov. And Mgmt. 10:112-125

Griffin, A (1997) PDMA research on new product development practices: Updating trends and benchmarking best practices. Journal of Prod. Innov. And Mgmt. 14:429-458

Hayes, W, and Zubrow, D (1995) Moving on up: data and experience doing CMM-based software process improvement. CMU/SEI-95-TR-008, Carnegie Mellon Software Engineering Institute

Kleiner, A and Roth, G (1997) How to make experience your company's best teacher. Harvard Business Review, September-October

Kuhn, D, Wells, C, Armitage, J, Cusick, K, Garcia, S, Hanna, M, Malpass, P, and Pierson, H (1996) A description of the systems engineering capability maturity model appraisal method version 1.1. CMU/SEI-96-HB-004, Carnergie Mellon Software Engineering Institute

March, JG (1994) A primer on decision making. New York: Free Press

Mercer Management Consulting, Inc. (1994) High performance New Product Development: Practices that set Leaders Apart. Boston: Mercer Management Consulting

Morbey, GK (1988) R&D: Its relationship with company performance. J. of Product Innovation Management 5: 191-200

National Research Council (1991) Improving engineering design: Designing for competitive advantage. NRC #PB91-208330

Olson, E, Walker, R, and Ruekert R (1995) Organizing for effective new product development: the moderating role of product innovativeness. J. of Marketing 59: 48-62

Paulk, MC, Curtis, B, Chrissis, MB, and Weber, CV (1993) Capability maturity model for software, Version 1.1. Software Engineering Institute Technical Report No. CMU/SEI-93-TR-24

Van de Ven, A, and D Polley (1992) Learning while innovating. Organization Science 3(1): 92-116

Williams, K (1997) The value of software improvement. SPIRE97, June 4

Item	Factor 1: Maturity	Factor 2: Project Outcomes
Documentation describes the product and its production processes	**0.68**	0.29
Project objectives include economic, market and product outcomes	**0.57**	0.19
Project planning emphasizes prevention of problems in projects	**0.78**	0.14
New products are developed using processes that are explicitly documented	**0.39**	0.24
Improving the NPD process is the responsibility of all project teams.	**0.65**	0.04
Improvement of the NPD process occurs through "Lessons Learned" disseminated across projects.	**0.65**	0.41
We try to prevent problems from occurring.	**0.71**	0.08
We try to control the develpment process through data on intermediate steps from multiple projects	**0.55**	0.43
Our projects were on schedule	0.15	**0.87**
Our projects were within budget	0.08	**0.88**
Project outcomes agree well with predicted expectations	0.44	**0.69**
Explained variance	**3.45**	**2.57**
Proportion of total	**0.31**	**0.23**
Cronbach's alpha (normalized)	**0.82**	**0.79**

Table 1 Final Scale Items

Table 2 Modeling Results

Term	Mean	Std.Dev.	Estimated Parameter (standardized)	t-value	p-value
Intercept			1.39	3.36	0.002
Maturity	3.02	1.11	0.74	2.09	0.045
Maturity—Company Size Interaction			-0.03	-0.46	0.646
Maturity—Market Volatility Interaction			-0.07	-0.85	0.401
Maturity—Industry Type Interaction			0.06	0.66	0.517

Adjusted $r^2 = 0.32$
Model p-value = 0.000287