Validating Determinants of Information Systems Development Maturity Model

Brian Salmans*

University of North Texas
College of Business
1155 Union Circle #311160
Denton, TX 76203-5017
940 453-8860
salmansb@unt.edu

Leon Kappelman

ITDS Dept P.O. Box 305249
University of North Texas
Denton, TX 76203-5249
940 565-3107
Leon.Kappelman@unt.edu

Robert Pavur

ITDS Dept P.O. Box 305249
University of North Texas
Denton, TX 76203-5249
940 565-3107
pavur@unt.edu

*Corresponding author

Formatted

Validating Determinants of Information Systems Development Maturity Model

ABSTRACT

The failure rate of information systems (IS) development projects has continued to plague organizations for decades. One response has been to implement a software process improvement program such as the Software Engineering Institute's Capability Maturity Model. Additionally, research such as Brooks (1995) and Jones (2008) indicate the failure to get requirements right is the most critical problem with IS development. Enterprise architecture (EA) may provide the discipline to bridge the gap between effective requirements, organizational objectives, and actual IS implementation. This research explores the relationship between IS development maturity and requirements practices informed by EA principles and investigates the impact of requirement determinants on IS development maturity.

Keywords: IS Development, Maturity Models, Requirements, Enterprise Architecture

1. INTRODUCTION

An organization's process maturity is of paramount importance to managing capabilities to meet vital requirements across functional areas within an organization. Carnegie Mellon's Software Engineering Institute developed a software Capability Maturity Model (SEI CMM) based on Humphrey's (1988) initial research on maturity models. The SEI CMM is the most widely known maturity model in the information technology (IT) arena (Rogoway, 1998), laying a foundation for the development of maturity models in a variety of business applications. The integration of an organization's business requirements and the design capabilities of an IS has a beneficial role in a process maturity model. Damian and Chisan (2006) state the relationship between requirements engineering and other IS development processes as "Requirements engineering is an important component of effective software engineering, yet more research is needed to demonstrate the benefit. While the existing literature suggests effective requirements engineering can lead to improvements,…there is little evidence to support this" (p. 433).

The purpose of this research is to examine the relationship between an organization's perceived position within the levels of a process maturity model and their position as determined by an adapted SEI CMM instrument. The standards within this instrument determine the maturity level of an organization. This research investigates the relationship between various requirements analysis and design (RA&D) efforts and activities and the self-reported and maturity levels of an IT organization. CIOs and mid-level IT Execs responded to a survey to assess their compliance with requirements analysis and design practices.

A logistic regression model considered high and low levels of both perceived and measured process maturity for each organization as the dependent variable and RA&D efforts and activities as the independent variables. The results of this analysis assist in establishing the important determinants of categorizing organizations into high and low levels of maturity. Descriptive statistics reveal an organization's perceived process maturity levels are substantially less than the measured levels of maturity. This difference may be due to organizations' conservative approach in assessing their progress in the development cycle. The logistic regression analysis provides insight into an understanding of the determinants that a CIO or mid-level IT executive uses, perhaps without realizing it, in classifying their organization into a specific maturity level of the process maturity model. These determinants are compared to those found using the levels of maturity measured by the adapted SEI CMM instrument.

This paper is organized by first providing background on the process maturity model and the importance of functional activities and services being aligned to enterprise objectives. Next, a method section presents the details of the analysis of variables thought to be related to an organization's choice of process maturity level. A discussion and conclusions section follows.

2. LITERATURE REVIEW

2.1 Process maturity models

Capability maturity models provide a basis for the control of IT and organizational processes and practices by identifying strengths, areas for improvement, and subsequent activities to effect improvement in the processes and practices. Humphrey (1988) defines a process "as a sequence of tasks that, when properly performed, produces the desired result" (p. 74). The standards defined by the maturity models establish levels of maturity and can be used in managing the IS or organizational improvements desired (Steghuis, Daneva, & van Eck, 2005). From Humphrey's work, Carnegie Mellon's Software Engineering Institute developed the software Capability Maturity Model (SEI CMM). It is used as a foundation in developing many other maturity models in a variety of applications and approaches (Rogoway, 1998). The primary goal of the SEI CMM is to improve IS development processes, practices and quality. The premise of the CMM is to better manage the software process (Paulk, Curtis, Chrissis, & Weber, 1993) and to address increasing quality and productivity issues in IS development (Duggan, 2004). A capability maturity model is any model which contains essential elements of effective processes and defines an improvement plan evolving to more mature processes, characterized by an improvement in quality and effectiveness (Chrissis, Konrad, & Shrum, 2003). A maturity model does not have to exist only for the IS discipline. The SEI CMM is not the only software process improvement model in existence. Other examples include the ISO/IEC 15504 standards and the software quality portion of ISO 9000. The SEI CMM contains five maturity levels, or stages, each comprising increasingly prioritized process goals. These five levels, in increasing level of maturity are: 1) initial, 2) repeatable, 3) defined, 4) managed, and 5) optimizing. The process goals help to prioritize IS development and the achievement of them evolutionarily increases the maturity of the IS development processes of the organization (Paulk et al., 1993).

Using the CMM to characterize and inform the IS development process maturity provides a strong basis for many reasons. One reason is the popularity and heavy usage of it which demonstrates a level of applicability and appropriateness (Beecham, Hall, & Rainer, 2005; El Emam & Madhavji, 1995). Indeed, Rogoway (1998) declares the CMM a *de facto* standard regarding process improvement assessments. Another reason is the adaptability of the CMM to the specific needs of the organization (Beecham, et al., 2005; Paulk et al., 1995). Reifer (2000) identifies 34 CMMs that have been developed. A third reason supporting the use of the CMM is the continued viability of it, the SEI continues to actively support it and update it (Beecham, et al., 2005).

2.2 Outcomes of SEI CMM

Prior research investigating the relationship between process maturity level and organizational outcomes has shown positive relationships between process maturity and software quality

(Harter & Slaughter, 2000), product quality (Bohn, 1995), and other increased quality outputs (Fenton & Neil, 1999; Ryan, 2000; Zahran, 1998). In early studies of CMM, Glass (1999) found a consistent trend of improvement when organizations implemented the CMM. An evolution in CMM level from 1 to 3 led to better software quality, higher investment returns, and higher productivity at Raytheon according to research conducted by Dion (1993). The improved IS development process and corresponding results at Raytheon also led to an increase in morale and a lowering of the absenteeism and attrition rates in the software development branch (Dion, 1993). A broader research survey of 138 individuals in 56 organizations also showed positive effects on staff morale from attaining higher CMM maturity (Goldenson, El Emam, Herbsleb, & Deephouse, 1996). Similarly, in a study of the effect of CMM on IS development, Clark (2000) found that an increase in maturity level of one can reduce the effort required in software development by between 4 to 11 percent. In a research study that combined the results of 12 case studies of 12 software development organizations, Hall, Beecham, and Rainer (2002) found the number of requirements problems tended to trend downward as the CMM level increased in these organizations. Finally, in a multi-year study of IS development practices, Jones (2002) concludes large applications (larger than 10,000 function points) tend to be more successful when the developing organizations are at or above level 3 of the CMM. Overall, organizations with a higher level of IS development maturity as measured by the SEI CMM experience an improvement in defect potentials, IS code with fewer high-severity defects, and increases in defect removal rates (Jones, 2008).

2.3 Information systems development

According to Brooks, the accidental part of software engineering is the coding and testing, the IS development aspects (Brooks, 1995; McConnell, 1999). Put another way, the language, tools, and methods in use for IS development are the accidents (Berry, 2003). The IS discipline has engaged many of the aspects or elements of IS development. Examples include increasingly more capable programming languages, development methodologies, and integrated environments (Brooks, 1995; McConnell, 1999). Advocating the tools aspects of IS development can be seen in the research by Beam, Palmer, and Sage (1987) who emphasize the combination of macroproductivity tools and microenhancement approaches to IS development. But one can see even with these improvements addressing many of the accidental elements, the state of IS development has not correspondingly improved. Projects still come in over budget and over time without getting the requirements right, the essence element of IS development. Brooks contends that attacking the essential problems of IS development is the only way for significant improvements.

IS development involves the “analysis, design, and implementation of applications and systems to support business operations in an organizational context” (Xia & Lee, 2005, p. 46). This is similar to Swanson and Beath’s (1989) definition of IS development as the “analysis, design, and implementation of new applications” (p. 296). Another conceptualization of IS development is to consider the end state or the goal sought. For example, the goal of IS development “is the effective creation of a set of work products, comprising an operational system and its supporting documents” (Wasserman & Freeman, 1983, p. 57). No matter the semantics used in framing IS development, the development of information systems is a complex endeavor in an environment of stiff competition, uncertainty, instability, and with frequent technological and market disruptions (Nidumolu & Subramani, 2003).

2.31 IS success and failures

However, even with the importance of IS development processes and outcomes to organizations, the record of IS development projects exhibits a high failure rate. Some researchers cite such statistics as 80 percent of IS projects are over budget and 25 percent are cancelled outright and of those that are not cancelled, 75 percent are operational failures, not operating as specified or simply not being used (Schmidt, Lyytinen, Keil, & Cule, 2001). The often-cited *Chaos Chronicles* report of the Standish Group International reports only a 34 percent IT project success rate in 13,522 projects at Fortune 500 firms (Nelson, 2005). A discouraging point is that the litany of IS development project ills, whether from cost or time overruns, functionality issues, or other such aspects begins the introduction sections of many research articles on IS development whether from the 1970s, 1980s, 1990s, or the twenty-first century (Duggan & Reichgelt, 2006). Many of these failures are the result of one or more of the following situations: total system failure, cost and/or time overruns, and implementation with reduced features or functions not meeting user requirements (Nelson, 2005; Yeo, 2002). A conservative estimate puts the annual cost of failed IS projects in the US from $60 to $ 70 billion annually (Charette, 2005). The consequences of these failures can result in decreased revenues, damage to corporate reputation, exposure to legal liabilities, and a decrease in productivity (Baltzan & Phillips, 2007). Moreover the failures of IS development can bring significant organizational consequences, even potentially leading to ruin (Goulielmos, 2003; Xia & Lee, 2004).

2.4 IS failure and requirements

There are many different reasons for these IS development problems, but the failure to get the system requirements right is considered to be one of the primary if not the most significant, reasons for this high IS failure rate (Duggan & Thachenkary, 2003. The failure of IS programs to get the requirements right has an equally lengthy history as IS development failures (Brooks, 1995). Indeed this is a continuing problem for the IS field and is the root cause for most IS project failures (Baltzan & Phillips, 2007). In his book *The Mythical Man-Month* Frederick Brooks (1995) discusses the essential importance of getting the requirements right in the software development process and the challenges the IS field has in getting those requirements right. Brooks (1995) states that "The hardest single part of building a software system is deciding precisely what to build. No other part of the conceptual work is as difficult as establishing the detailed technical requirements…. No other part of the work so cripples the system if done wrong. No other part is more difficult to rectify later." (p. 199). This difficulty in getting the requirements right was echoed decades later by Cheng and Atlee (2007) who maintain the requirements process is difficult because requirements are in the problem space, not the solution space like other software artifacts.

2.5 Requirements analysis and design

The critical role of requirements in IS development can be understood in the context of Brooks' (1995) accident vs essence hypothesis. Brooks asserts the difficulties of IS development can be framed with difficulties of essence and difficulties of accident. Generally, improvements in IS development have come from resolving the accidental difficulties, those that served as barriers making software tasks difficult. But Brooks maintains resolving the essential difficulties--difficulties inherent in the general nature of software--should be the focus because it would bring about order-of-magnitude improvements in IS development. These inherent difficulties of IS development derive from software's essential properties of complexity, conformity, changeability, and invisibility (Brooks, 1996, McConnell, 1999; Xia & Lee, 2005) and continue to be relevant today (Duggan & Reichgelt, 2006). Brooks (1995) concludes, "Many of the classical problems of developing software products derive from this essential complexity" (p. 183). In a similar tone, Sawyer, Sommerville, and Viller (1998) state: "No software process, whatever its 'capability', can keep delivery times, costs and product quality under control if the requirements are poorly formulated or unstable" (p. 1).

2.51 The requirements process

Whether or not IS development projects are successful depends on how successfully they meet the needs of the applicable users and the usage environment (Cheng & Atlee, 2007; Nuseibeh & Easterbrook, 2000). These needs are the requirements. Beecham et al. (2005) define the requirements process as "activities performed in the requirements phase that culminate in producing a document containing the software requirements specification" (p. 248). Generally, the process of gathering requirements consists of the following core functions: 1) eliciting requirements; 2) modeling and analyzing requirements; 3) the communication of requirements; 4) agreement on the requirements; and 5) evolving or changing the requirements (Nuseibeh & Easterbrook, 2000). The process of garnering requirements from the stakeholders and users of a prospective IS is of critical importance to IS development, and is one of the most (if not *the* most) difficult steps in IS development, and has effects throughout the system development life cycle (Browne & Rogich, 2001; Halbleib, 2004). A successful requirements process will increase the probability of successful information systems development (Browne & Rogich, 2001; Davis, 1982). Even with the aforementioned studies indicating the benefits to IS development from the initiation of software performance improvement programs such as CMM, there remains a significant and critical neglect of research into the relationship of requirements and IS development (Sawyer, Sommerville, & Viller, 1998). As Sommerville and Ransom (2005) put it, requirements are mostly outside the scope of process improvement models.

2.52 Requirements and the IS Development Lifecycle

To understand the importance of requirements to the entire IS development lifecycle, it is useful to view it from the context of IS development errors. For instance, discovering errors during the requirements phase costs significantly less to fix then finding the errors during the IS development or implementation phases (Berry, 2003; Jones, 2006). Some estimates put an exponential growth pattern with the cost of errors in IS development, with an error in the the requirements stage being $5 while one at the construction or actual production stage being $100 and $1000 respectively (Hay, 2003).

Another approach to understanding the importance of requirements analysis and design to IS development is research by Jones (2008) into measuring defect potentials. Jones (2008) defines defect potentials as "the probable numbers of defects that will be found during the development of software applications" (p. 2). Based on his study of around 600 organizations and 13,000 IS projects, Jones maintains the most serious IS project defects are not actually in the code but in the requirements and design and that in the US, these defect potentials average 45 percent of all software defects. This growth in costs of errors throughout the system development life cycle, specifically within the requirements phase manifests itself with research indicating an increased and improved effort in the conduct of requirements analysis and design leads to a better implementation of information systems (Berry, 2003; Daugulis, 2000).

2.6 Enterprise architecture as a context for requirements

The critical relationship between IS requirements and information systems development has only increased in importance as IS investments continue to account for significant slices of organizational budgets and as software's role in supporting strategic business goals becomes more important (Harter & Slaughter, 2000). These information system investments are increasingly being reviewed to ensure alignment with and contribution to organizational goals and strategies. Additionally, the trend of more outsourcing, strategic alliances, and demands for more security and privacy safeguards within information systems creates an atmosphere demanding more in depth knowledge of not only IS requirements but also an in depth knowledge of the organization itself (McNurlin & Sprague, 2006).

Enterprise architecture facilitates the alignment between the business and IT domains, a possible answer to achieving conceptual integrity and getting the requirements right and dealing with the requirements challenges highlighted by Brooks. Enterprise architecture may be the discipline that can bridge the gap between the integration of organizational objectives and requirements with the actual implementation through IS development. To put it in the context of Brooks' (1995) hypothesis, EA may be the effort that can best attack the essential, not the accidental difficulties in software development and regarding IS requirements. It facilitates the attainment of a comprehensive view of enterprise-wide requirements (Kappelman, McGinnis, Pettit, Salmans, & Sidorova, 2008). Enterprise architecture ensures congruency between organizational strategies, processes, and IS requirements, forming an inclusive IS strategy (Young, 2001). Thus a central goal of EA is the alignment of IS requirements to the goals and objectives of an organization (Henderson & Venkatraman, 1993).

Enterprise architecture provides a formalized way to capture and document an organization's present and future desired state and thus contributes to the management of change to the desired state. As an emerging discipline, there is still not a consensus on the definition of EA. According to one of the "founders" of EA, it is "that set of design artifacts, or descriptive representations, that are relevant for describing an object such that it can be produced to requirements (quality) as well as maintained over the period of its useful life (change)" (Zachman, 1997, p. 44). Another source defines EA as the "analysis and documentation of an enterprise in its current and future states from an integrated strategy, business, and technology perspective" (Bernard, 2005, p. 31). In other words, it provides a formalized way to capture and document an organization's present and future desired state of requirements and thus can contribute to the management of change to the desired state of requirements. Finally, EA is defined by Ross, Weill, and Robertson (2006) as "the organizing logic for core business processes and IT infrastructure reflecting the standardization and integration of a company's operating model" (p. viii). The discipline of EA has been increasingly applied in organizations in order to facilitate a disciplined approach to policy, planning, decision-making, and resource development. However, even with the increasing strategic importance of EA in globalized countries (Ross, Weill, & Robertson, 2006; Stutz & Schelp, 2007) there has not been enough basic research into EA, which adversely affects the maturity of it (Langenberg & Wegmann, 2004).

2.7 The research gap between IS development and requirements

A gap remains between theory and practice in the phenomena of IS system failures (Yeo, 2002). Even though there has been much research on the role (and deficiencies) of requirements in IS development and management, and the outcomes of software process assessment and improvement programs such as the SEI CMM, there is little research regarding the nature of the relationship between IS development and requirements analysis. As discussed by Damian and Chisan (2006) in research into the relationships between requirements engineering and other IS development process, "Requirements engineering is an important component of effective software engineering, yet more research is needed to demonstrate the benefits…While the existing literature suggests effective requirements engineering can lead to improvements…there is little evidence to support this" (p. 433). Organizations typically embark on a software process improvement program (eg. SEI CMM) either for compliance or to address problems with their IS development programs such as poor software quality, functionality issues, or over time and budget (Layman, 2005). However, there is a gap in accounting for the vital role of requirements within the software process improvement programs. As mentioned previously, there is substantial research describing and indicating positive outcomes from the SEI CMM and this research is validated and based on the use of SEI CMM in practice. This, however, is not the case with requirements. Though research indicates positive outcomes of improved requirements on IS development, adoption and use by practice has not occurred as it has with the CMM.

3. ANALYSIS OF THE DETERMINANTS OF LEVELS OF PROCESS MATURITY

The purpose of this research, therefore, is to examine the relationship between IS development maturity and RA&D practices. This question is framed in the larger context of enterprise architecture. For this research, the relationship between IT development maturity and RA&D is displayed in Figure 1. Regarding the levels of IS development maturity (CMM), few organizations will claim to be at the highest (Optimized) level as completion of all critical success factors is difficult. Descriptively, the perceived level within this model from a survey of 376 IT professionals revealed the following categorizations for levels 1 through 5, respectively: 108, 140, 72, 43, 2. Obviously, there is a steep decline in the number of professionals reporting their perception of an organization's maturity level after the maturity level of 2. Twelve questions from the adapted SEI CMM instrument reveals the measured maturity level to be more optimistic as indicated by the following categorizations for levels 1 through 5, respectively: 3, 62, 159, 144, and 8. This divergence in frequencies may simply reflect the conservative nature of professionals when being so close to their organizations' improvement process that it seems much lower than that indicated after tallying the critical factors actually completed.

Figure 1. Model used associating RA&D with process maturity model.

IS Development Maturity

Requirements Analysis & Design Practices

The following hypothesis will be addressed using logistic regression analysis between categorical high and low variables for the IT Development Maturity construct and variables most representative of RA&D practices. The most representative variables are selected from a step down logistic regression procedure using 15 variables in the full model.

H1: Higher IS development maturity is associated with higher RA&D practices.

4. METHOD

In order to better understand the relationship between IT development maturity and requirements within an EA context, a survey was conducted under the auspices of the Society for Information Management Enterprise Architecture Working Group (SIMEAWG). Preliminary definitions of EA functions and benefits were developed by the authors based on existing EA literature, including EA maturity models as well as EA and IT-business alignment research (e.g., Luftman & McLean, 2004; Luftman, 2007; van der Raadt, Hoorn, & van Vliet, 2005; van der Raadt, Soetendal, Perdeck, & van Vliet, 2004). Enterprise architecture maturity models were integrated as a foundational structure and to inform the requirements analysis and design items within the survey. A variety of EA maturity models were reviewed, with a core of four EA maturity models used for instrument development. Additionally, the key IT and business alignment enablers and inhibitors posited by Luftman and McLean (2004) were integrated into the survey to reflect the importance of alignment to EA. The definitions of EA functions and benefits were further refined through a modified Delphi study approach with an expert group of EA professionals from industry and academia. The information systems design maturity items were derived from the SEI CMM. These had been used in a previous survey by Kappelman (1997) regarding the Y2K problem and Y2K strategies. The respondents were asked to express their agreement or disagreement with the suggested statements on a 5-point Likert scale.

The survey was initially distributed in a pilot test to the SIMEAWG with the intent for this expert group to develop final recommendations and revisions for the survey. The final survey was comprised of 80 questions of which 14 were demographic questions. An invitation to participate in the survey with an embedded hyperlink to the web-based survey was sent out by e-mail to individuals on SIM's membership list. Measures were taken so that each participant could not participate in a survey more than once. Over the subsequent six weeks, three reminder e-mails were sent to individuals who had not completed the survey. To encourage accurate responses, each potential participant had an option to include an e-mail address to receive a report of the research findings. A total of 2863 surveys were sent, with about 376 quality responses after data purification.

5. RESULTS

A total of 376 quality responses were received, which represents a 13.13 percent response rate, this is consistent with other surveys of SIM members. Analysis of demographic data showed 42.2 percent of respondents were C-level executives, Directors represented 32.2 percent. 'Other' roles accounted for 21 percent. Nearly two-thirds of respondents (63 percent) had enterprise-wide responsibilities, 16 percent had business unit responsibilities, while the remaining 21 percent had departmental or team responsibilities. A logistic regression model was analyzed with the dependent variable being high and low perceived process maturity and the independent variables were RA&D measures/practices (15 items). A total of 353 useable responses were divided into high and low process maturity based on the perception of being at level 3 or higher (110 professionals) or less than maturity level 3 (243 professionals). A step-down logistic regression procedure with an entry and keep significance level of 20% yielded 5 variables out of the 15 RA&D variables as being important determinants with 1 variable being significant at the 10% level and 3 at the 5% level. The significance levels are reported in Table 1.

Table 1. (Available upon request.)

A similar logistic regression model was analyzed with the dependent variable being high and low process maturity as measured by the adapted SEI CMM instrument (12 items) and the independent variables were the same RA&D measures as in the first logistic regression model. The same 353 respondents were used to categorize their organization's process maturity levels into high and low. Because of the higher values obtained on the scale versus the perceived levels, a median value of 3.5 was used to split the group into high and low maturity levels. In the process maturity levels, there were 105 professionals in the low level and 248 professionals in the high level. A similar step-down logistic regression procedure yielded the results in Table 2.

Table 2. (Available upon request.)

6. DISCUSSION AND CONCLUSION

The results in Tables 1 and 2 support hypothesis H1 since key RA&D practices were found to significantly support the classification of high and low levels of process maturity both for perceived levels and levels determined by the split using SEI CMM maturity values. These findings indicate the requirements practices that help to distinguish the higher or lower levels of maturity of an organization's IS development processes. The inclusion of the requirements variables (RA&D efforts and activities…): 18a labeled as "are measured", 18b labeled as "are benchmarked", and 18d labeled as "highly developed and disciplined" in both logistic regression models demonstrates organizational RA&D practices that are consistent with the language found in the higher information systems development maturity stages or levels within the SEI CMM. These findings are strong indications validating Brooks' (1995) notion that requirements are the essence of the IS development process and that we must focus on the essential difficulties, not the accidental ones. Higher levels of maturity in requirements practices contribute to higher levels of IS development maturity which can lead to many of the benefits experienced by organizations with higher levels of IS development maturity, such as lowering defect potential rates (and lowering the associated costs of defects in the IS development life cycle) (Jones, 2008).

When using the 12 variable, adapted SEI CMM instrument to provide an indication of an organization's IS development maturity, several additional RA&D practices emerge demonstrating their contribution to higher levels of IS development maturity. These additional practices may not be considered by IT professionals when providing a snapshot assessment of their IS development process maturity. The exclusion of the variable (RA&D efforts and activities are…) 18k labeled as "are viewed strictly as an IT initiative" from this same model may indicate the awareness of the IT professionals that requirements must transcend organizational functional areas and cannot be IT-centric to satisfy organizational objectives with the information systems being developed.

7. REFERENCES (Available upon request)