Association for Information Systems
AIS Electronic Library (AISeL)

ICIS 2000 Proceedings

International Conference on Information Systems (ICIS)

December 2000

Process Maturity and Software Quality: A Field Study

Donald Harter
University of Michigan

Sandra Slaughter
Carnegie Mellon University

PROCESS MATURITY AND SOFTWARE QUALITY: A FIELD STUDY

Donald E. Harter
University of Michigan Business School
U.S.A.

Sandra A. Slaughter
Graduate School of Industrial Administration
Carnegie Mellon University
U.S.A.

Extended Abstract

Quality has emerged as a key issue in the development and deployment of software products (Haag et al. 1996; Prahalad and Krishnan 1999; Yourdon 1992). As software products play an increasingly critical role in supporting strategic business initiatives, it is important that these products function correctly and according to users' specifications. The costs of poor software quality (in terms of reduced productivity, downtime, customer dissatisfaction, and injury) can be enormous. For example, the Help Desk Institute, an industry group based in Denver, estimates that in 1999, Americans spent 65 million minutes on "hold" waiting for help from software vendors in debugging software problems (Minasi 2000).

An unresolved issue is how software quality can be improved. On the one hand, some software researchers and experts argue that quality can be tested into software products. That is, defective software products can eventually become "bug-free" through rigorous testing (Anthes 1997; Hanna 1995). However, on the other hand, there is a notion that quality must be designed or built into software products from the start (Fenton and Neil 1999). That is, quality in design predicts quality in later stages of the product life cycle. There is some empirical evidence to support this notion from Japanese software factories (Cusumano 1991), the NASA Space Shuttle program (Keller 1992), and a variety of projects at IBM (Buck and Robbins 1984).

If the level of quality persists throughout a product's life cycle, how can quality be designed into the product? In manufacturing, Bohn (1995) found evidence that process maturity (i.e., the sophistication, consistency, and effectiveness of manufacturing processes) was positively associated with product quality. This relationship is believed to exist because as a process becomes more mature and less variable, the outputs of the process (i.e., products) have a higher level of quality (Fenton and Neil 1999; Ryan 2000; Zahran 1998). In the context of software production, this implies that maturity of the software development process is essential to reducing process variability and thus improving the quality of software products (Humphrey 1988).

There is some empirical support linking process maturity to software quality. For example, Diaz and Sligo (1997) found initial evidence of a positive relationship between process maturity and software quality at Motorola. Herbsleb et al. (1997) found additional anecdotal support of this relationship, but suggest that further research is needed to understand more precisely how process maturity and software quality are related. Determining whether process maturity is linked to software quality is important because, in practice, many managers still emphasize testing at the end of the development cycle instead of building in quality through better processes (Anthes 1997; Hanna 1995).

Thus, this study has been designed to address the following question that is central to these issues: *What is the relationship between process maturity and software quality over the product life cycle*? We develop a conceptual framework (Figure 1) for assessing the relationship between process maturity and software quality at different stages of the product life cycle: development, implementation, and production. Our models are empirically evaluated using archival data collected on software products developed over 12 years by the systems integration division of an information technology company. Based upon our analysis, we identify the direct and indirect marginal effects of improved process maturity on software quality at the different stages of the software life cycle. Our results also provide insight into the question of whether quality is a persistent characteristic of software

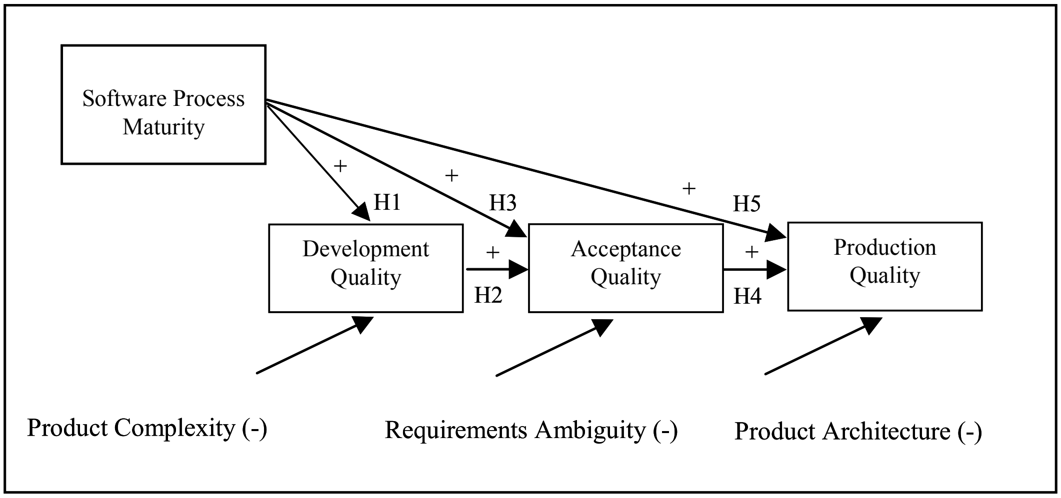

Figure 1. Software Quality Conceptual Model

Table 1. Development Quality Parameter Estimates (n = 33)
(standard errors, t statistics, and one-tailed p values)

$$\textit{Development-Quality} = \alpha_{01} + \alpha_{11} * \textit{Process-Maturity} + \alpha_{21} * \textit{Product-Complexity} + \varepsilon_{Q1}$$

Variable	Parameter	Weighted OLS Estimate	Weighted SURE Estimate
Intercept	α_{01}	**2511.690**	**2862.554**
	s.e.	1129.927	1086.158
	t	2.224	2.635
	p	0.017	0.004
Process-Maturity	α_{11}	**896.801**	**917.741**
	s.e.	219.241	206.464
	t	4.090	4.445
	p	0.000	0.000
Product-Complexity	α_{21}	**-972.220**	**-1095.455**
	s.e.	402.439	382.682
	t	-2.416	-2.863
	p	0.011	0.002
R^2		**0.364**	**0.403**
R^2 (adj)		**0.322**	
F Model	**$F_{2,30}$**	**8.580**	
	P	**0.001**	
χ^2	**$\chi^2(2)$**		**21.127**
	P		**0.000**

Table 2. Acceptance Quality Parameter Estimates (n = 31)
(standard errors, t statistics, and one-tailed p values)

$$\textit{Acceptance-Quality} = \alpha_{02} + \alpha_{12} * \textit{Process-Maturity} + \alpha_{22}\ \textit{Requirements-Ambiguity} + \alpha_{32} * \textit{Development-Complexity} + \varepsilon_{Q2}$$

Variable	Parameter	Weighted OLS Estimate	Weighted SURE Estimate
Intercept	α_{02}	**-1523.423**	**-1509.442**
	s.e.	1857.695	1663.113
	t	-0.820	-0.908
	p	0.210	0.182
Process-Maturity	α_{12}	**937.406**	**1248.846**
	s.e.	453.215	424.607
	t	2.068	2.941
	p	0.024	0.002
Requirements-Ambiguity	α_{22}	**94.438**	**32.316**
	s.e.	542.381	487.126
	t	0.174	0.066
	p	0.432	0.474
Development-Quality	α_{32}	**1.099**	**0.834**
	s.e.	0.348	0.321
	t	3.162	2.600
	p	0.002	0.005
R^2		**0.495**	**0.526**
R^2 (adj)		**0.439**	
F Model	**$F_{3,27}$**	**8.830**	
	P	**0.000**	
χ^2	**$\chi^2(3)$**		**31.455**
	P		**0.000**

products, i.e., is quality designed into or tested into software products. We conclude by discussing the contributions of our work and the implications of our findings for software quality research and practice.

The analysis indicates that a higher level of process maturity leads to higher software quality. Quality at each stage increases as a direct benefit of higher maturity, but also improves due to quality built into the product at earlier stages (see Tables 1, 2, and 3). This second point suggests the persistent nature of quality, reinforcing the belief that quality can be designed into products but not necessarily tested into products at later stages.

Our study makes two primary contributions to the literature on software quality and process improvement. First, we have developed and empirically developed a conceptual framework for assessing the inter-relationships between software quality and process maturity throughout the product life cycle. Second, our findings provide insight about the nature of software quality and the impact of process maturity on software quality. Our results suggest that software quality is designed into products rather than tested into products; a high quality software product in development has high quality in acceptance and later in production. Correspondingly, software products with poor design have low quality that persists throughout the software life cycle. Further, we find that process maturity has a positive benefit on software quality at each stage of the life cycle and that the persistent nature of quality magnifies the effect of process maturity on quality as products proceed through the life cycle.

Defects are generated throughout the life cycle. Defects generated in one phase can go undetected until much later. It appears that improved process maturity inhibits the introduction of defects early in the life cycle. By eliminating these defects, the process

Table 3. Production Quality Parameter Estimates (n = 32)
(standard errors, t statistics, and one-tailed p values)

$$\textit{Production-Quality} = \alpha_{03} + \alpha_{13} * \textit{Process-Maturity} + \alpha_{23}\ \textit{Product-Architecture} + \alpha_{33} * \textit{Acceptance-Quality} + \varepsilon_3$$

Variable	Parameter	Weighted OLS Estimate	Weighted SURE Estimate
Intercept	α_{03}	**3759.621**	**4028.903**
	s.e.	5951.563	6001.567
	t	0.632	0.671
	p	0.267	0.251
Process-Maturity	α_{13}	**4079.357**	**4414.103**
	s.e.	2518.279	2453.556
	t	1.620	1.799
	p	0.058	0.036
Product-Architecture	α_{23}	**-2922.052**	**-3023.189**
	s.e.	2190.578	2138.786
	t	-1.334	-1.414
	p	0.097	0.079
Acceptance-Quality	α_{33}	**2.200**	**1.903**
	s.e.	0.877	0.841
	t	2.510	2.261
	p	0.009	0.012
R^2		**0.481**	**0.502**
R^2 (adj)		**0.426**	
F Model	$F_{3,27}$	**8.660**	
	P	**0.000**	
χ^2	$\chi^2(3)$		**24.929**
	P		**0.000**

prevents them from being propagated through latter stages. At the same time, process maturity is directly reducing defects in these latter stages. The effect of reduction of defects in each life cycle phase, combined with the reduction in defects carried forward from previous phases, magnifies the overall quality improvement through a cascading effect of process maturity on quality. From an economic perspective, this is extremely important since errors detected later in the life cycle are substantially more expensive to correct.

References

Anthes, G. H. "Quality? What's That?," *Computerworld* (31:41), October 13, 1997, pp. 75-76.

Bohn, R. E. "Noise and learning in semiconductor manufacturing," *Management Science* (41:1), January 1995, pp. 31-42.

Buck, R. D., and Robbins, J. H. "Application of Software Inspection Methodology in Design and Code," *Software Validation*, H. L. Hausen (ed.), Amsterdam: Elsevier Science, 1984, pp. 41-56.

Cusumano, M. A. *Japan's Software Factories*, Oxford: Oxford University Press, 1991.

Diaz, M., and Sligo, J. "How Software Process Improvement Helped Motorola," *IEEE Software* (14:5), September 1997, pp. 75-81.

Fenton, N. E., and Neil, M. "A Critique of Software Defect Prediction Models," *IEEE Transactions on Software Engineering* (25:5), September/October 1999, pp. 675-689.

Haag, S., Raja, M. K., and Schkade, L. L. "Quality Function Deployment Usage in Software Development," *Communications of the ACM* (39:1), January 1996, p. 41.

Hanna, M. "Test Early, Test Often," *Software Magazine* (15:10), October 1995, pp. 59-63.

Herbsleb, J., Zubrow, D., Goldenson, D., Hayes, W., and Paulk, M. "Software Quality and the Capability Maturity Model," *Communications of the ACM* (40:6), June 1997, pp. 30-40.

Humphrey, W. S. "Characterizing the Software Process: A Maturity Framework," *IEEE Software* (5:2) March/April 1988, pp. 73-79.

Keller, T. "Measurements Role in Providing Error-Free Onboard Shuttle Software," in *Proceedings of the Third International Applications of Software Metrics Conference*, Software Quality Engineering, La Jolla, California, 1992 pp. 2.154-2.166.

Minasi, M. "Software Industry's Bugs Unacceptable," *Houston Chronicle*, Houston, TX, January 30, 2000 p. 5.

Prahalad, C. K., and Krishnan, M. S. "The New Meaning of Quality in the Information Age," *Harvard Business Review* (77:5) September/October 1999, pp. 109-118.

Ryan, T. P. *Statistical Methods for Quality Improvement,* New York: John Wiley & Sons, 2000.

Yourdon, E. *Decline and Fall of the American Programmer*, Englewood Cliffs, NJ: . Prentice Hall, 1992.

Zahran, S. *Software Process Improvement: Practical Guidelines for Business Success*, Essex, England: Addison Wesley Longman Ltd., 1988.