Association for Information Systems
AIS Electronic Library (AISeL)

ICIS 2003 Proceedings

International Conference on Information Systems (ICIS)

December 2003

The Impact of Schedule Pressure on Software Development: A Behavioral Perspective

Ning Nan
University of Michigan

Donald Harter
Syracuse University

Tara Thomas
University of Michigan

THE IMPACT OF SCHEDULE PRESSURE ON SOFTWARE DEVELOPMENT: A BEHAVIORAL PERSPECTIVE

Ning Nan
University of Michigan
Ann Arbor, MI USA
nnan@umich.edu

Donald E. Harter
Syracuse University
Syracuse, NY USA
dharter@syr.edu

Tara Thomas
University of Michigan
Ann Arbor, MI USA
taramt@umich.edu

Abstract

Timely software development has been a major issue in both information systems research and software industry. While researchers and practitioners seek better techniques to estimate and manage software schedules, it is important to understand the impact of management pressure on software development projects. This paper investigates the impact of schedule pressure on the performance in software projects. Data analysis indicates that a U-shaped function exists between time pressure and cycle time. A similar relationship is found between time pressure and development effort. Meanwhile, time pressure does not significantly affect software quality. The findings of this study will help software project managers develop effective deadline and budget setting policies.

Keywords: Software development estimation, schedule pressure, IS development time, IS development effort, software quality

Research Objective

Software development has been a critical topic in the information systems field. While most research attention has been paid to estimation and performance, a few studies have looked at the impact of schedule pressure on software development (e.g, Abdel-Hamid 1988, 1989; Abdel-Hamid et al. 1993). However, when examining the issue of pressure, most research neglected the behavioral dimension. As Gutierrez and Kouvelis (1991, p. 990) pointed out, "ignoring behavioral issues in modeling project activity durations is equivalent to assuming that there is no relationship between the actual amount of work to be done, the deadline set for the worker to finish that work, and the actual completion time of the work." Therefore, the objective of this paper is to examine the impact of schedule pressure from a behavioral perspective. The thesis of this paper is that management and clients create schedule pressure on software developers, which in turn impacts the software development processes and outcomes. We attempt to answer the following questions:

1. What is the impact of schedule pressure on software development cycle time?
2. What is the impact of schedule pressure on software development effort?
3. What is the impact of schedule pressure on software quality?

Answers to these questions will help both researchers and practitioners understand the relationship between pressure and job performance. With this knowledge, project managers can leverage the potential positive effects of pressure and enhance the performance and productivity of their work force.

Literature Review

Behavioral scientists have extensively studied the relationship between pressure and task performance. Some psychologists regard pressure as a major element producing stress. Since stress often means a negative response to the environment, they believe that greater pressure usually leads to worse performance. In contrast, Parkinson (1957) believed that work would expand so as to fill the time available for its completion (Parkinson's Law). Parkinson's Law implies that greater pressure should lead to higher productivity and better performance. The two seemingly contradictory views are reconciled by a U-shaped relation proposed by other researchers. These researchers argue that the linear relations seen by previous research are linear fractions of a large U-shaped curve.

The behavioral perspectives on pressure and performance have been implicitly or explicitly applied to several studies on IT management. The COnstructive COst MOdel (COCOMO) (Boehm 1981) incorporate schedule constraints as one of its 15 cost drivers. It assumes that any deviation from a nominal schedule, either acceleration or stretch out, will incur more development effort. The effort multipliers of COCOMO imply a V-shaped relation between schedule constraints and development effort. That is, development effort is linearly increasing with either negative or positive deviation from the nominal schedule. This V-shaped relation approximates the U-shaped function proposed by behavioral scientists. Consistent with the behavioral theories, COCOMO suggests that the lowest amount effort occurs under an intermediate level of schedule constraints. Although Boehm discussed the possibility of a nonlinear relationship between schedule constraints and development effort, he did not explore the exact curvilinear shape. In addition, Boehm did not explicitly define schedule pressure in his study. Nor did he explore the impact of schedule pressure on actual cycle time, development effort, and software quality.

Abdel-Hamid and his colleagues (Abdel-Hamid 1988, 1989; Abdel-Hamid and Madnick 1989, Abdel-Hamid et al. 1993; Abdel-Hamid et al. 1999) constructed a system dynamics model to study software development processes. In this model, schedule pressure is a factor of error rate, management decision making, and project productivity. Results from a series of experimental simulations suggest that schedule pressure has significant impacts on cycle time and development effort. The system dynamics model provides a holistic view of the connections between schedule pressure and other elements of a software development project. However, it does not specify the function form of the relations between schedule pressure and other elements in software development processes.

In a recent study, Austin (2001) examined the effects of time pressure on quality in a software engineering context. The relationship between pressure and software development was formulated into a principal-agent game model. In the game model, penalties perceived by the agent decided payoff. Furthermore, the probability that an agent is assigned a task deadline that is not achievable without taking quality-compromising shortcuts determined the pure Nash equilibrium. Although the model left out important behavioral factors such as motivation and stress, its findings have important implications to this paper. According to the analytical solution of the game model, subtracting time from estimates so that few developers can regularly meet the deadlines creates the optimal deadline-setting policy. When a developer sees that most of his/her colleagues cannot meet a deadline, he/she feels less obligated to meet the deadline. As a result, the developer are more likely to take time to improve the quality of a project.

In summary, both the behavioral and the management literature indicate that pressure created by the discrepancy between the ideal estimated schedule and the management constrained schedule has significant impact on job performance and productivity. Particularly, behavioral scientists predict a U-shaped relation between pressure and performance. IT management researchers have seen some indications of this U-shaped relation in software engineering projects. Based on the related literature, this paper hypothesizes that

> *H1: Pressure created by the discrepancy between the ideal estimated time and a management constrained deadline has a significant impact on software development cycle time, with the actual cycle time increasing for projects whose pressure is either very high or very low.*

> *H2: Pressure created by the discrepancy between the ideal estimated time and a management constrained deadline has a significant impact on software development effort, with the actual effort increasing for projects whose pressure is either very high or very low.*

Although schedule pressure affects software development cycle time and effort, previous research suggests that it may not have a significant impact on quality. Both COCOMO and Abdel-Hamid's studies show that when schedule pressure is low, developers will spend more time and effort on planning rather than quality improvement. Meanwhile, Austin's study shows that when schedule pressure is high, developers will increase work time and effort to maintain the quality of their work. Hence, this paper proposes that:

> *H3: Pressure created by the discrepancy between the ideal estimated time and a management constrained deadline does not significantly affect software quality.*

Although previous research has touched on the issue of schedule pressure, no one has used the behavioral theories to explain and predict the impact and potential benefits of schedule pressure in a software engineering context. The hypotheses of this paper can lead to more insights into the relationship between schedule pressure and software development outcomes. In addition, this paper helps software managers to understand the behavioral mechanism behind the relationship.

Research Methodology

Data Collection

Data examined in this paper was collected between 1984 and 1994 from a $1 billion/year information technology firm. Process improvement data were collected from government auditors and personnel in external divisions. These independent groups used the Software Engineering Institute's capability maturity model (Paulk et al. 1995) to assess the maturity of the IT firm's software development and supporting activities.

Construct Measurements

Time Pressure

Although time pressure has been extensively studied by behavioral researchers, no clear definition or measurement exists for it in the literature. Time pressure for software developers usually results from varied schedule rather than changed workload. Due to the ambiguity of the definition of time pressure in the literature, we developed our own measurement of time pressure:

$$\text{Time Pressure} = \frac{\text{Team Estimated Cycle Time} - \text{Management Shortened Cycle Time}}{\text{Team Estimated Cycle Time}}$$

Since customers seldom add slack to the estimation, this paper assumes that values for time pressure are positive and within the range between 0 and 1.

Cycle Time and Development Effort

In software engineering research (e.g., Harter et al. 2000), cycle time usually measures the number of calendar days elapsed from the first day of design to final customer acceptance of the product. The total number of person months logged by the development team in all stages of product development starting from initial design through final product acceptance testing constitutes the development effort. Previous research indicated that both cycle time and development effort are highly correlated with product size. Therefore, in this paper, these two measurements are normalized by product size to eliminate the size effect.

Product Quality

The definition and measure of product quality are based on previous software engineering research. Specifically, it is measured as the number of lines of source code in the product divided by the sum of defects found in system and acceptance testing (Harter et al. 2000). This measure is the inverse of the defect density used in many previous quality studies (e.g., Fenton and Pfleeger 1997). This paper adopts the inversed defect density because it offers a more intuitive understanding of the quality values.

Analysis and Results

Regression Models

The general function forms for the data analysis are:

Normalized Cycle Time = *f*(process, complexity, quality, time pressure)
Normalized Effort = *f*(process, complexity, quality, time pressure)
Quality = *f*(process, size, complexity, time pressure)

As suggested by previous studies (e.g., Banker and Kemerer 1989), one potential method to test a U-shaped relationship is to add a quadratic term as an independent variable. This paper thus specifies the statistical models as:

$$\begin{aligned} \textit{Normalized Cycle Time} = {} & \beta_{01} + \beta_{11}\,(\textit{process}) + \beta_{21}\,(\textit{complexity}) + \beta_{31}\,(\textit{quality}) + \\ & \beta_{41}\,(\textit{time pressure}) + \beta_{51}\,(\textit{time pressure})^2 \end{aligned} \tag{1}$$

$$\begin{aligned} \textit{Normalized Effort} = {} & \beta_{02} + \beta_{12}\,(\textit{process}) + \beta_{22}\,(\textit{complexity}) + \beta_{32}\,(\textit{quality}) + \\ & \beta_{42}\,(\textit{time pressure}) + \beta_{52}\,(\textit{time pressure})^2 \end{aligned} \tag{2}$$

$$\begin{aligned} \textit{Quality} = {} & \beta_{03} + \beta_{13}\,(\textit{process}) + \beta_{23}\,(\textit{size}) + \beta_{33}\,(\textit{complexity}) + \\ & \beta_{43}\,(\textit{time pressure}) + \beta 53\,(\textit{time pressure})^2 \end{aligned} \tag{3}$$

Previous studies indicated that process maturity, product size, and product design complexity have significant impacts on cycle time, development effort, and product quality (e.g., Harter et al. 2000). Therefore, in the regression models, these variables are tested as control variables. We adopt the measures of these control variables from the relevant literature. Process maturity is measured by the Software Engineering Institute's CMM level of maturity. Product size is measured by KLOC (thousand lines of source code). Product design complexity is the average score of domain, data, and decision complexity.

Results

The models were initially tested by using ordinary least squares (OLS), but because data in this study are all from the same company, it may be possible that the error terms are correlated as a result of some common effect. Therefore, we also estimated the seemingly unrelated regression (SURE) parameters using a feasible generalized least squares (FGLS).

Generally speaking, OLS and SURE produced consistent results. Both OLS and SURE found positive and significant coefficient of the quadratic terms in the cycle time equation (Equation 1). Both indicate that the quadratic term in the quality equation (Equation 3) is not significantly different from zero. The only difference is that OLS did not produce a significant coefficient of the quadratic term in the effort equation (Equation 2) while SURE did. In sum, SURE estimates support for all the hypotheses whereas OLS only verified hypotheses 1 and 3.

We tested the assumptions of the OLS and SURE estimators. A Shapiro-Wilk test (Shapiro and Wilk 1965) did not reject the assumption of normality for any of the models at the 5 percent significance level. White's (1980) test verified homoskedasticity of residuals in all of the models. In addition, we conducted Breusch-Pagan test on the SURE residuals. It confirmed the homoskedasticity of the SURE estimator errors. Using criteria established by Cook and Weisberg (1982), we identified and removed one influential outlier.

Since our models include a linear term and a quadratic term of schedule pressure, we would expect a high degree of multicollinearity between the two variables, confirmed by high variance inflation factors. Using a technique proposed by Banker et al. (1993), we tested the robustness of the model with highly correlated linear and quadratic terms. These results are consistent with the estimates from the original models.

Discussion

Since SURE accounted for the potential correlation among disturbances in our models and produced consistent results with OLS, we use the SURE estimates for the interpretation of our results.

In the cycle time equation (Equation 1), we find that schedule pressure has a significant impact on software development cycle time, with the actual cycle time increasing for projects whose pressure is either very high or very low ($\beta = 139.462$, $p < 0.01$). By differentiating cycle time with respect to schedule pressure, we get 0.55 as the optimal schedule pressure value. That is, the shortest cycle time occurs when management reduces the team estimated cycle time by 55 percent. The positive coefficient of the quadratic term means that the actual cycle time rises at an increasing rate when schedule pressure gets farther away from the optimal point.

Our findings of the effort model (Equation 2) show that schedule pressure affects software development effort, with the actual effort increasing for projects whose pressure is either very high or very low ($\beta = 6.934$, $p < 0.05$). We differentiated effort with respect to schedule pressure and found 0.74 as the optimal pressure value. It means that when management shortens the team estimated cycle time by 74 percent, the project will take the least effort. As is in the cycle time model, the coefficient of the quadratic term is positive. This indicates that when schedule pressure gets farther away from the optimal value, the actual effort will increase at a growing rate.

In the quality model (Equation 3), we found that the coefficients of the linear and quadratic pressure terms are not significantly different from zero. It suggests that schedule pressure does not substantially affect quality of the software. This result may be counterintuitive, since we often assume that people make more mistakes under greater pressure. Boehm (1981) pointed out that when schedule pressure is low, software management tends to allocate more time and effort to the initial planning. Abdel-Hamid et al. (1999) argue that "initial project plans do not necessarily constitute the best course of action to take during the project's life cycle" (p. 533). In other words, under low schedule pressure, the increased time and effort are often unrelated to quality improvement. On the other hand, when schedule pressure is high, developers tend to spend more time and effort to maintain quality rather than meet deadlines.

When interpreting the above results, we are aware of the small sample size behind them. Sample size is one of the three parameters defining the power of any statistical test. Small sample size can impair the power of statistical tests. Low statistical power often prevents researchers from identifying an existing effect. In MIS research, the average level of statistical power is relatively low (Baroudi & Orlikowski 1989). A major reason is that it is very difficult to collect a large amount of empirical data. Our study, like many other MIS studies, is constrained by the availability of data points. However, with such a small data set, we could find statistically significant effects of schedule pressure in all three models. This indicates that the effect size of schedule pressure is very strong.

Implications and Contributions

In this paper, we introduced the relevant behavioral theories to examine the important issue of estimation in software development. In addition, we developed an intuitive and effective measurement of time pressure in the context of software engineering management.

The data analysis reveals several interesting results. First, adding time pressure to software development teams can shorten the actual development time. Managers can increase the productivity of their teams by imposing tighter schedules. However, there is a limit to this beneficial effect of pressure. When management cuts too much time off a team estimated schedule, developers will burn out and end up taking more time to finish their work.

Second, schedule pressure can not only shorten cycle time, but also reduce development effort. The reduced cycle time under schedule pressure is not paid off by an increased work force. On the contrary, with a properly tightened schedule, software management can save time and manpower. However, there is a boundary to this positive effect.

Finally, when schedule pressure reduces cycle time and effort, it does not affect software quality. To software managers, balancing cost and quality is an important issue. Our results imply that project managers can maintain quality while applying schedule pressure to reduce cost. In other words, under the optimal level of pressure, software development teams can finish projects in shorter time, with less effort, and of good quality.

In the highly competitive software engineering industry, companies are looking for ways to make software production more efficient. The results of this study can help project managers understand the relationship between pressure, job performance, and software quality. Such understanding can lead to more effective deadline and budget setting policies, which ultimately add to the competitive advantage of an IT company. Future research can explore factors leading to the optimal level of pressure and productivity.

Reference

Abdel-Hamid, T. K "The Dynamics of Software Projects Staffing: A System Dynamics Based Simulation Approach." *IEEE Transactions on Software Engineering* (15:2), February 1989, pp. 109-119.

Abdel-Hamid, T. K "The Economics of Software Quality Assurance: A Simulation-Based Case Study," *MIS Quarterly* (12:3), September 1988, pp. 395-411.

Abdel-Hamid, T. K., and Madnick, S. E. "Lessons Learned from Modeling the Dynamics of Software Development," *Communications of the ACM* (32:12), December 1989, pp.1426-1455.

Abdel-Hamid, T. K., Sengupta, K., and Ronan., D. "Software Project Control: An Experimental Investigation of Judgment with Fallible Information" *IEEE Transactions on Software Engineering* (19:6), June 1993, pp. 603-612.

Abdel-Hamid, T. K., Sengupta, K., and Swett, C. "The Impact of Goals on Software Project Management: An Experimental Investigation" *MIS Quarterly* (23:4), December 1999, pp. 531-555.

Austin, R. D. "The Effects of Time Pressure on Quality in Software Development: An Agency Model," *Information Systems Research* (12:2), June 2001, pp. 195-207.

Banker, R. D., and Kemerer, C. F. "Scale Economies in New Software Development," *IEEE Transactions on Software Engineering* (15:10), October 1989, pp. 1199-1205.

Banker, R. D., Datar, S, M., Kemerer, C. F., and Zweig, D. "Software Complexity and Maintenance Costs," *Communications of ACM* (36:11), November 1993, pp. 81-94.

Baroudi, J. J., and Orlikowski, W. J. "The Problem of Statistical Power in MIS Research," *MIS Quarterly* (13:1), March 1989, pp. 87-106.

Boehm, B. W. *Software Engineering Economics,* Prentice-Hall, Inc., Englewood Cliffs, NJ, 1981.

Cook, R. D., and Weisberg, S. *Residuals and Influence in Regression*, Chapman & Hall, London, 1982.

Fenton, N. E., and Pfleeger, S. L. *Software Metrics: A Rigorous and Practical Approach*, International Thompson Computer Press, London, 1997.

Gutierrez, G., and Kouvelis, P. "Parkinson's Law and its Implications for Project Management," *Management Science* (37:8), August 1991, pp. 990-1001.

Harter, D. E., Krishnan, M. S., and Slaughter, S. A. "Effects of Process Maturity on Quality, Cycle Time, and Effort in Software Product Development," *Management Science* (46:4), April 2000, pp. 451-466.

Parkinson, C. N. *Parkinson's Law and Other Studies in Administration*, Random House Inc., New York, 1957.

Paulk, M. C., Weber, C. V., Curtis, B, and Chrissis, M. B. *The Capability Maturity Model: Guidelines for Improving the Software Process*, Addison-Wesley, Reading, MA, 1995.

Shapiro, S., and Wilk, M. "An Analysis of Variance Test for Normality," *Biometrika* (52), 1965, pp. 591-612.

White, H. "Heteroskedasticity-Consistent Covariance Matrix Estimator and a Direct Test for Heteroskedasticity," *Econometrica* (48:5), 1980, pp. 817-838.