REDUCING SOFTWARE PRODUCT DEVELOPMENT TIME

by

John Callahan

Carleton University

and

Brian Moretton

Ottawa-Carleton Police

Send correspondence to:

Professor J. Callahan

School of Business, Carleton University,

Ottawa, Ontario, Canada K1S 5B6

phone: (613) 788-2372, fax: (613) 749-7960

Email: john_callahan@carleton.ca

September, 1998

REDUCING SOFTWARE PRODUCT DEVELOPMENT TIME

Abstract

This research explores project management approaches for reducing software product development times. Data were gathered on 44 software product development projects and used to test a set of seven hypotheses. Generally, we found that the newness of the product type for the firm affects which approaches are effective. Low experience and high experience projects are different. For low experience projects, supplier involvement and sales and marketing involvement early in the development cycle were found to lead to shorter development time. Also supported was the use of frequent load builds and system tests of the whole product. There was little evidence for the influence of these approaches for high experience projects.

Suggested keywords

Software product; Development time; Management approaches

Introduction

Product development time is of increasing importance, and this importance is reflected in the significant amount of research that has been done on the determinants of product development time. This research, however, has been carried out using data on projects for the development of physical products. Software products are different as are the technical and organizational processes used to develop them.

Software is pure design--the laws of physics are not constraints. Manufacturing is not an important issue. Software design is “in the code” so that it is not visible making it hard to use software design as a focal point for development project coordination and integration, unlike many physical designs which can be made visible to all project participants. The re-use of design is not as far developed in software although there are serious efforts in this direction. These differences mean, at the very least, that the determinants of product development time so far elicited from research on physical products may not apply for software products. There may be other factors which uniquely affect development time. Carmel [1], in an exploratory study of software product development time, found developers of PC application software generally unaware of development time reduction as a management concept. Developers tended to focus on rapid development, with an emphasis on “crunch” periods of intense effort aimed at meeting a deadline.

The importance of product development time is well documented [2, 3, 4, 5, 6]. Shortened product life cycles have meant that short development time is even more critical [3]. Short development time allows a competitor to beat others to the market. Being late to market, i.e., arriving with a product after other competitors have entered the market successfully, means significantly lower revenues and dramatically lower profits [7]. Short development time also provides a firm with considerable flexibility. If a market window for a particular type of product is forecast to open up in 24 months, and a firm can develop the product for the market in 15 months while its competition takes 24, then the firm can afford to wait for further market intelligence and technology development whereas its competition must begin serious development immediately.

Although the evidence is compelling that development time is important, there are some who are critical of an overemphasis on it as a management variable. Ittner and Larcker [8], for example, find little evidence that faster cycle time *alone* increases firm performance. Burchill and Fine [9] found that, during product concept development early in the development cycle, an overemphasis on reducing time to market decreases concept development time but inadvertently causes delays and detours in downstream development activities and increases total development time. Regardless, development time remains an important management concern.

This paper addresses the lack of research on software product development time. We use the research literature on development time of physical products as a base to investigate the determinants of software product development time. We find that product newness affects the determinants of software development time. Strong evidence is found that supplier involvement and sales and marketing involvement early in the development cycle leads to shorter development time. Also strongly supported is the use of frequent load builds. There was also evidence in support of system tests of the whole product as being effective in reducing product development time.

The paper is structured as follows. In the next section, we review the literature on the determinants of software development time. In the following section, we develop a set of seven testable hypotheses. We then describe the research design: data gathered and variables measured. After presenting the results of our analysis of the data, we discuss the results and put them into the context of the research literature. In the last section, we present the principal conclusions of the paper.

Research on Product Development Time

We use four recent journal articles as the basis for our review of the research literature on the determinants of product development time: Zirger and Hartley [10], Datar et. al. [11], Griffin [3], and

Eisenhardt and Tabrizi [12], none of which are cross referenced as the studies were carried out at about the same time.

Zirger and Hartley [10] used a sample developed from 44 usable questionnaires collected from 120 general managers of firms in the electronics industry participating in a 2 week executive course. As expected, they found that increasing the number of functions on a development team, having a dedicated team, setting and measuring fast cycle time as an explicit project goal, and overlapping development activities all contributed to decreasing development time. Unexpectedly they found that decreasing the number of major suppliers tended to increase development time and, very surprisingly, that projects with higher levels of management support were in fact slower. Factors found not significant were: degree of product change, number of parts, freezing product design early, degree of part design conducted by suppliers, co-location, team authority.

Datar et al [11] used a sample of 220 new products for which complete histories on time to prototype, time to volume production, and engineering expenditures were available from three manufacturers which dominate a section of the electronic components industry. They explored the relationship between development project structure and time to market. They defined a concentrated new product development structure as one in which all product designers are in one facility. A concentrated structure facilitates cross product learning but may not provide proximity to customers and production engineers. In contrast, a distributed structure disperses new product development effort among multiple manufacturing locations. Their major finding was that *time to volume production* is shorter in a distributed structure than in a concentrated one. Perhaps even more importantly, they found the impact of their explanatory variables to be highly *nonlinear*.

Griffin [3] compiled a sample of 274 completed and 69 ongoing projects (343 in total) from 21 divisions of 11 companies in five industries. Her sample included both product and service development projects. She found that the number of months spent in product development increases with higher product complexity and higher product newness. As a result, she suggests that researchers control for these variables. She also found that formal product development processes decrease cycle time more for complex products, and that cross-functional teams decrease cycle time more for projects with high newness.

Eisenhardt and Tabrizi [12] contrasted two strategies that enable firms to achieve fast adaptation through product innovation: compression and experiential. They proposed the compression strategy to be most appropriate when product development is certain and predictable and can be planned out as a series of steps. The process can then be accelerated by shortening the time of each step, overlapping successive development stages, and rewarding developers for attaining the compressed schedule. They proposed the experiential strategy most appropriate when product development is uncertain. In this case the keys to fast development are being able to learn quickly about changing conditions and remaining flexible in order to adapt to these changes.

Eisenhardt and Tabrizi [12] associated the following determinants of product development time with the compression strategy: planning, supplier involvement, Computer Aided Design (CAD) usage, project overlap, multifunctional teams, and rewards for schedule attainment. They associated multiple iterations, frequent milestones, testing and project leader power with an experiential strategy for shortening development time. Using data on a sample of 72 product development projects gathered as part of a larger data gathering exercise in 36 large PC, minicomputer, mainframe and peripheral companies, they found support for their proposition regarding the two types of strategies for reducing product development times. Specifically, their results supported the link between fast product development and multifunctional teams and the experiential strategy of iterations, testing, milestones, and powerful leaders. Their results for the influence of CAD usage, supplier involvement and stage overlap were mixed. They found that planning and rewards for schedule attainment were associated with slow product development.

Carmel [1] notes the lack of research literature on reducing software development time. Our thesis is that several of the determinants identified in this literature for shortening product development time can be adapted and applied to the reduction of the development time of software products. We now elaborate this contention in the form of several testable hypotheses.

Determinants of Software Development Time

Planning

The arguments for the beneficial influence of planning on product development time are natural ones. Planning should allow a product development team to gain a better understanding of product and project requirements early in the development process. Hayes, Wheelwright and Clark [13] noted the importance of pre-development planning for fast product development. Time spent in planning should shorten product development time by rationalizing the process, reducing mistakes and shortening delays [14, 15, 16]. Ancona and Caldwell [17] found, however, that it was possible for product development teams to engage in too much planning and to diminish team performance as a result. Eisenhardt and Tabrizi [12] argue that planning can be particularly effective for projects involving the development of products whose characteristics are familiar to the developing firm, but less so when the development project involves significant leaning.

Planning the development of a software product development is different--if only for the inherent differences in software as a product. Manufacturing is not a critical consideration, for example. Software is malleable and its design tends to be a less structured process than that for physical design. Thus the influence of planning on development time may be less significant. Regardless, the relationship bears testing.

These consideration lead us to the hypothesis that:

Hypothesis 1: More time spent planning is associated with shorter software development time.

Supplier Involvement

The arguments for the efficacy of supplier involvement in reducing product development time are straightforward as well, and many studies have tested the relationship. Delegation of the execution of some development steps to supplier organizations can reduce the workload of the focal team allowing it to focus on the execution of tasks that take advantage of its skills and competencies [18, 19]. Suppliers bring focused design expertise to the table as well. Mabert, Muth, and Schmenner [20] found that suppliers were involved in many stages of fast projects. The nature of supplier involvement which contributes to product development has also been investigated by Hartley et. al. [21] who concluded that key suppliers should be involved as early as possible and significant benefits can be gained by selectively shifting design to suppliers. On the other hand, the development of industry wide modular and reusable design in software has lagged other industries [22]. Standardized off-the-shelf software components are not common. This may lead to a less important role for suppliers of software components in the software product development process. Design tool providers are also suppliers in the software context.

These considerations lead us to test the hypothesis that:

Hypothesis 2: More supplier involvement is associated with shorter software development time.

Sales and Marketing Involvement

Multifunctional teams are closely linked to successful product development since different development stages rely on different functional expertise [19]. These teams allow technical, marketing, and manufacturing activities to be integrated and permit downstream problems in functional areas to be recognized earlier in the process when they are easier to fix [18]. Involving more functions early in a development project can also reduce wait time [18]. However, most research in the area of functional involvement in product development has been driven by considerations of early involvement by manufacturing to develop manufacturing process concurrently to product design. As mentioned above, manufacturing is not a significant consideration in software product development. However, incorporating customer requirements into software product design remains important so that the involvement of sales and marketing in software product development should also.

These considerations lead to the hypothesis that:

Hypothesis 3 (H3): Sales and marketing involvement early in the development process is associated with shorter software development time.

Rewards

Rewards tied to meeting a schedule would seem to have the potential of focusing the attention of developers on the project at hand and on time-based performance rather than on other issues such as the technological sophistication of the product, although Eisenhardt and Tabrizi [12] found that such rewards were associated with slow product development. They attributed this result to reverse causality: that firms experiencing slow product development resort to higher levels of reward for schedule attainment.

Rewards for schedule attainment may not work well in software development which is often genuinely iterative. What has actually been attained in an iterative process is hard to measure, and when outputs are hard to measure economic theory tells us that work inputs are more logically used to set compensation.

We still considered the relationship worth investigating. Thus:

Hypothesis 4: Greater financial rewards for schedule attainment are associated with shorter software development time.

Build Frequency

It is argued that multiple design iterations accelerate the product development process by facilitating learning and improving the ability of developers to recognize the requirement for a shift in direction [23, 12]. The spiral model of development is increasingly influential in the software industry [24]. It is an iterative approach: the design process cycles through the stages of specification development, high level design, detailed design and testing.

A load build is a system level compilation of a software product that software engineers utilize to test the product and find errors [25]. It is common to have many load builds during a software development project. When developing a radically new product, more frequent load builds offers rapid feedback to the development team about how the product is progressing. Each build allows developers to gain a better understanding of the product [25]. The frequency of these builds depends on the particular needs of a project and the amount of time required to complete a build successfully.

Using a load build as a proxy for design iteration in the sense tested by Eisenhardt and Tabrizi [12] leads to the hypotheses that:

Hypothesis 5: More frequent load builds are associated with shorter software development time.

Testing

Testing can be used to discover design problems earlier in the process when they are easier to rectify [15]. This is particularly important in uncertain environments. Testing can also reduce development time because the data gathered provides a basis for improving the product. Three important methods of testing software products during development are unit tests, system tests, and beta tests. Unit tests are used by developers to discover and fix bugs while coding. System tests are internal, laboratory type tests of versions of the whole product. Beta tests occur when the product is tested by an end user/customer often on their site.

Thus we have the following hypothesis:

Hypothesis 6: Testing is associated with shorter software development time.

Project Leader Power

Project leaders can have a critical effect on the speed and productivity of the development process [19]. These leaders assist in accelerating product development by keeping the process focused, and by being able to obtain the co-operation of a wide range of disparate interests within a firm and to co-ordinate their inputs. However, because of the differences between software and physical product development alluded to above (particularly the lack of manufacturing issues), it is not clear whether or not the importance of project leader power which has been demonstrated in the development of physical product development holds for software.

Thus we have:

Hypothesis 7: Greater project leader power is associated with shorter development time.

Research Design

The unit of analysis for this study is the software development project. Email requests to participate in the study were sent to the 107 Canadian and American software product development companies which: (a) were listed in one of three databases: Industry Canada Stategis (http://strategis.ic.gc.ca), CanadaIT (http://www.canadaIT.com) and Tickit (http://www.tickit.com); (b) which were headquartered in North America; and (c) had over 20 employees. Fifty-five product development project leaders of these companies agreed to complete the questionnaire. Questionnaires were sent out and returned via email. Forty-four usable questionnaires were returned for a response rate of 44/107 or 41%. The response rate for Canadian companies was 25/63 or 40%; for American companies, it was 19/44 or 44%.

The largest firm in the Canadian sub-sample had 1997 revenues just over $250 (Cda) million. The largest firm in the American sub-sample had 1997 revenues of about $150 (US) million. The size distribution of the sample firms was highly skewed with just over 80% of the sample firms having revenues of less than $2 million.

Respondents were project leaders in charge of software development projects. They were selected as respondents because they were knowledgeable about the project and are also responsible for project control.

Eisenhardt and Tabrizi [12] divided development projects into five stages: predevelopment, conceptual design, product design, testing, process development, and production start-up. However, because of the iterative nature of software development processes [24], it is difficult to determine in what “stage” a software development project is at any given time. Therefore, we moved away from dividing the development process into stages. Rather, we identified five major activities of software development which are common to all development processes and may be undertaken concurrently:

- defining product requirements--the determination of the commercial (or customer) requirements for the product together with its functional specification;
- system design--the definition of the product design at the level of the modules to be used and their interdependencies;
- creation of source code--the creation of the source code for the software product in whatever languages that are used;
- system testing--the internal, laboratory based testing of versions of the entire product; and
- beta testing--the testing of pre-release versions of the product by an end user/customer often at one of their sites.

Independent variables

Planning was measured by the proportion of total development time spent defining product requirements prior to the start of coding.

Supplier involvement was calculated as the sum of the rated importance of supplier personnel in the different activities of product development activities (among the five identified above) in which they participated. Participation was defined as having one or more employees of a supplier as recognized members of the product development team, actively participating in team meetings. Respondents rated on a 7-point scale the importance of supplier personnel in making key decisions during their participation (1 = Not Important, 7 = Very Important). For example, if in a given project, supplier personnel participated in defining product requirements, system design and system testing but not in creation of source code or beta testing, and their importance in making key decisions was rated as 5, 3 and 1 in the three activities in which they participated, the supplier involvement measure would be 5+3+1 = 9.

Sales and marketing involvement was measured by asking whether personnel from sales and marketing were involved on the development team during the five activities of product development identified above. Involvement was defined as having one or more functional employees as recognized members of the product development team, actively participating in team meetings.

Schedule reward was restricted to monetary rewards. It was measured by the proportion of total base pay for a project formed by schedule attainment bonuses.

Build frequency was found by dividing total number of load builds which were carried out during development of a product by total development time.

Testing was measured three ways. The respondent was asked to indicate how many times the system was tested as a whole product and how many beta test sites there were for the product. A third measure was developed by asking the respondent to rate the emphasis place on unit testing to the overall testing of the product.

Project manager power was measured as the extent of control that project leader respondents felt that they had over product specifications and over annual evaluations of development personnel.

Control variables

As suggested by Griffin [3], prior **experience** was used as a control variable. It was measured by asking the respondent to rate on a 7-point Likert scale the company's prior experience developing such a product. The anchors were 1-none and 7-considerable.

Cost was used to control for the size of projects. It was measured using the square root total project cost.

Dependent variable

Duration, the dependent variable, was measured using the square root of the interval between the point in time at which personnel were first assigned to analyze software product requirements and the release of the product to customers.

Results

Table 1 presents the Spearman correlations for the study variables. The variables are labeled with the numbers of the hypotheses to which they relate.

The control variable, cost, is strongly correlated with the dependent variable, duration. Four of the dependent variables, supplier involvement, sales and marketing involvement in defining product requirements, number of system tests, and project leader power over annual evaluations are correlated with duration--each with the anticipated sign providing some evidence in support of hypotheses 2, 3, 6 and 7.

Some other significant correlations warrant comment. Number of system tests and build frequency are positively correlated. This would seem natural. As would number of system tests being negatively correlated with emphasis on unit testing; and project leader power over product specification and over annual evaluations being positively correlated. The number of betasites is

Table 1: Spearman Correlations for the Study Variables

(two tailed, whole sample, n=44)

	D	**C**	**1**	**2**	**3**	**4**	**5**	**6**			**7**
								a)	b)	c)	a)
D Duration											
C Cost	.63***										
1 Planning	-.12	-.06									
2 Supplier involvement	-.28*	.03	.12								
3 Sales and mktg involv't in defining prod. req.	-.36**	-.09	.20	.16							
4 Schedule reward	.01	.12	.21	.25*	-.04						
5 Build frequency	-.13	-.02	-.19	.12	.12	-.04					
6 Testing											
a) emphasis on unit testing	.09	.10	.24	-.16	-.04	-.00	-.22				
b) number of system tests	-.33**	-.29*	-.16	.08	.18	.02	.38***	-.27*			
c) number of betasites	-.23	-.03	-.09	.07	-.04	-.30**	-.08	-.07	.04		
7 Project leader power											
a) over product specification	-.25*	-.35**	.12	-.16	.03	.09	-.11	.03	.17	.11	
b) over annual evaluations	-.33**	-.26*	.07	.21	.21	.09	-.04	-.01	.10	-.03	.37**

*p<.10, **p<.05; ***p<.01

negatively correlated with schedule reward, and positively correlated with supplier involvement--these reasons for these correlations are less obvious.

Table 2 presents correlations between the independent variables and both duration and cost for the whole sample (n=44) as in Table 1, and for two sub-samples: low experience (n=21) and high experience (n=23). Respondents were asked to rate their organization's prior experience in developing a product like the one which anchored their responses to the questionnaire on a 7-point Likert scale. Low experience projects were those with scores of 4 or less. High experience projects were those with scores of 5 or greater.

Note that the correlation results for the two sub-samples are very different. None of the independent variables are significantly correlated with duration in the high experience sub-sample. On the other hand for the low experience sub-sample, supplier involvement, sales and marketing involvement in defining product requirements, build frequency, number of system tests, number of betasites and project leader power over product specification all were significantly negatively correlated with duration--providing support for hypotheses 2, 3, 5, 6 and 7.

Table 2: Spearman Correlations for the Independent Variables Against the Dependent and Control Variables, Duration and Cost (two tailed)

	Full sample (n=44)		**Low experience (n=21)**		**High experience (n=23)**	
	Duration	**Cost**	**Duration**	**Cost**	**Duration**	**Cost**
1 Planning	-.12	-.06	-.13	-.34	-.08	.04
2 Supplier involvement	-.28*	.03	-.38*	-.16	-.13	.12
3 Sales and marketing involvement in defining product requirements	-.36**	-.09	-.48**	-.26	-.22	.05
4 Schedule reward	.01	.12	-.16	-.09	.28	.35
5 Build frequency	-.13	-.02	-.38*	-.22	.32	.10
6 Testing						
a) emphasis on unit testing	.09	.10	.29	.34	-.05	-.06
b) number of system tests	-.33**	-.29*	-.44**	-.62***	-.17	-.03
c) number of betasites	-.23	-.03	-.38*	-.27	-.07	.11
7 Project leader power						
a) over product specification	-.25*	-.35**	-.41*	-.35	-.11	-.30
b) over annual evaluations	-.33**	-.26*	-.32	-.27	-.25	-.35*

*p<.10, **p<.05; ***p<.01

The differences between the two sub-samples is further illustrated in Table 3 which presents correlations between sales and marketing involvement in defining product requirements and the other independent variables.

Table 3: Correlations between Independent Variables and Sales and Marketing Involvement in Defining Product Requirements (two tailed)

	Low experience (n=21)	**High experience (n=23)**
	Sales and marketing involvement in defining product requirements	**Sales and marketing involvement in defining product requirements**
1 Planning	.06	.39*
2 Supplier involvement	.38*	-.04
4 Schedule reward	-.14	.06
5 Build frequency	.61***	-.42
6 Testing		
a) emphasis on unit testing	-.62***	.53***
b) number of system tests	.42**	-.05
c) number of betasites	.22	-.26
7 Project leader power		
a) over product specification	.30	-.17
b) over annual evaluations	.09	.03

*p<.10, **p<.05; ***p<.01

Low experience projects which involved sales and marketing in defining product requirements also involved suppliers, emphasized build frequency and system tests while de-emphasizing unit testing. High experience projects which involved sales and marketing in defining product requirements, and emphasized planning and unit testing.

Stepwise regression analysis was used to develop optimum predictor models for software development time as a function of the independent variables. Table 4 shows the results for the full sample, and for the low and high experience sub-samples.

In the whole sample, supplier involvement, sales and marketing involvement in defining product requirements, and build frequency significantly affect duration. Again, results for the two sub-samples are different. For the low experience sub-sample, only sales and marketing involvement in defining product requirements is significant. For the high experience sub-sample, only supplier involvement is significant. All of the coefficients are of the hypothesized sign, lending support to hypotheses 2, 3 and 5.

Table 4 - Results from Stepwise Regression Analysis

	Full Sample (n=44)	**Low experience (n=21)**	**High experience (n=23)**
Adjusted R-square	.55	.60	.42
F (significance)	14.19***	15.98***	8.84***
C Cost	5.75***	4.23***	3.91***
1 Planning			
2 Supplier involvement	-2.59**		-1.98*
3 Sales and marketing involvement in defining product requirements	-3.53***	-2.94***	
4 Schedule reward			
5 Build frequency	-2.04**		
6 Testing a) emphasis on unit testing b) number of system tests c) number of betasites			
7 Project leader power a) over product specification b) over annual evaluations			

*p<.10, **p<.05; ***p<.01

Summary of the Results

1. The first hypothesis was that more time spent planning is associated with shorter software development time. Planning was measured by the proportion of total development time spent defining product requirements prior to the start of coding. There was no support for the hypothesis in the data. Total time spent defining product requirements prior to the start of coding was correlated with total development time as would be expected, but as a proportion of total development time it is not.

2. The second hypothesis was that more supplier involvement is associated with shorter software development time. There is considerable support for this hypothesis. Supplier involvement is significantly negatively correlated with duration. It is also a significant variable in regressions for the whole sample and for the high experience sub-sample. It would seem that supplier involvement is more significant as a determinant of development time for projects about which more is known at the start of development.

The effects on duration of supplier involvement in the five major activities of software product development (defining product requirements, system design, creation of source code, system testing and beta testing) was also investigated. Supplier involvement in each of these activities was used in place of overall supplier involvement in the regression analyses reported in Table 4.

Supplier involvement in defining product requirements, system design and beta testing all entered their regressions as significant determinants of product development duration. Supplier involvement in creation of source code and system testing did not. Thus supplier involvement both early in the development cycle, and late, contributed to short development times.

The significance of supplier involvement in the five major activities was different for the low and high experience sub-samples. Supplier involvement in defining product requirements was significant in both the low ($p<.10$) and high ($p<.05$) experience sub-samples. Supplier involvement in system design was not significant for the two sub-samples. Supplier involvement in beta testing was significant in the high experience sub-sample ($p<.05$) but not for the low.

3. Hypothesis three is that sales and marketing involvement early in the development process is associated with shorter software development time. Again, there is considerable evidence to support this hypothesis. Sales and marketing involvement in defining product requirements is negatively correlated with duration, and is a significant explanatory factor in the regression analyses, for both the whole sample and for the low experience sub-sample. Sales and marketing involvement in the other four major activities of software development which generally come later in the development cycle was also measured. There was no significant correlations between sales and marketing involvement in these activities and development time.

4. Hypothesis four was that greater financial rewards for schedule attainment are associated with shorter software development time. There was no evidence in the data to support this hypothesis.

5. That more frequent load builds are associated with shorter software development time was the fifth hypothesis. Build frequency was significantly negatively correlated with development time in the low experience sub-sample. It was also a significant explanatory factor in the regression analysis for the whole sample. Thus there is some support in the data for the hypothesis, particularly for low experience projects.

6. The sixth hypothesis was that testing is associated with shorter software development time. There is some support for this hypothesis in the data. The number of system tests was negatively correlated with project duration in both the whole sample and in the low experience sub-sample. Number of system tests did not enter the stepwise regressions as a significant determinant of development time, but it was significant when regressed against duration with cost as a control variable. Note, however, that number of system tests was negatively correlated with cost. That is, the bigger a project the fewer tests of the system as a *whole* product. Emphasis on unit testing was not significant. Number of betasites was significantly correlated with duration with the hypothesized sign in the low experience sub-sample, but nowhere else.

7. Hypothesis 7 was that greater project leader power is associated with shorter development time. Again, there is little evidence in support of this hypothesis in the data for the same reasons as those for hypothesis 6. Project manager power in the sample, as measured, tends to be negatively correlated with project size. Project managers tend to have a lower degree of control over product specification and annual evaluations of personnel in larger projects.

Discussion of the Results

There are a number of similarities and differences between the results of this research and those of previous studies on the development of "physical" product. Similar in general to the work of Eisenhardt and Tabrizi [12] and Griffin [3] was the fact that previous firm experience in developing like products was a significant factor in the influence of the determinants of development time.

The lack of effect of planning on development time is not that surprising given the fact that Eisenhardt and Tabrizi [12] found planning to be associated with slow product development. It may well be that the extent of "planning" is not captured well by using the time spent in a development project before some pre-defined cutoff point such as the definition of product requirements. This is particularly true for software development in which iterative, "spiral" development is becoming the dominant approach.

While in general, the results for supplier involvement support Imai et. al.'s [18] contention that supplier involvement will expedite development time and in particular Hartley et al.'s [21] finding that early supplier involvement is particularly effective, they contradict Eisenhardt and Tabrizi's [12] hypothesis that supplier involvement will shorten development time for experienced companies. This suggests that the role of suppliers in software development may differ from the role of suppliers in other industries. Although firms do contract the creation of separate modules that comprise a software product, more often than not, the supplier relationship with software companies entails the provision of development tools. In his response to the questionnaire, one project leader stated that, "with more knowledge about what (their) suppliers are doing such as products they are working on and their scheduled release dates, (they) would have improved our product development." Since these tools are constantly changing and improving, maintaining a close relationship with suppliers can be important. This may be particularly true for low experience projects because managers may not be aware of the best development tools to use or may not have the resources to develop their own tools. Further exploratory research is required along the line of that of Hartley et. al. [21] on the nature of suppliers' contribution to software product development.

The strong evidence in support of early sales and marketing involvement is congruent with many studies of the development of physical products. The lack of effect of such involvement in activities occurring later in the development cycle is remarkable.

Contrary to their expectations, Eisenhardt and Tabrizi [12] found a positive relationship between reward for schedule attainment and long development times. They ascribed this relationship to inverse causality, arguing that when a project is over long managers would tend to start rewarding personnel for schedule. In a sense, the lack of a relationship in this research is a better result. It should be noted, however, that the data for reward for schedule which was restricted to monetary rewards was highly skewed. Only 14 of the sample of 44 reported any such rewards at all.

As noted above, load builds were used to capture the iterative nature of a design process in a way analogous to the use of design iterations by Eisenhardt and Tabrizi [12]. Their finding that design iterations are significantly related to shorter development time for high uncertainty projects is analogous to our finding that load builds are significant determinants of development time particularly for low experience projects.

There is some support for the role of system testing in reducing development time, however, after analyzing the data one is left with the conviction that with better constructs significant relationships along the lines hypothesized would have been supported. Eisenhardt and Tabrizi [12] used total time spent testing and level of report of development project manager as their measures of testing and project manager power. They found support for their hypotheses. Entering the research, it was felt that number of system tests and betasites would be more direct measures of testing. Total time spent testing was also not attractive--it was felt that respondents would have a very difficult time estimating this given the kind of iterative parallel development typical of software product development.

The results for project manager power were disappointing. Analysis of the data on project manager power as measured: control over product specification and over annual personnel evaluations, leads us conclude that other measures are also needed to capture this construct. Required are constructs not so directly related to project size as was ours. Bowen et. al. [26] provide a list of “powerful manager” attributes. We might have used “direct contact with the customer”, for example, from their list.

Two open-ended questions were asked at the end of the questionnaire: over which aspects of their development process would project leaders like more control, and how would they improve their development processes. Responses to these questions provide a basis for further investigation into project leader power. The aspect which is most interesting about these questions is that areas of development over which project leaders would like more control were not necessarily areas which they would improve, and vice versa. For example, project leaders indicated that they would improve product testing, but they did not indicate that they would like more control over that aspect of the development process. The same applies for training of development personnel. Project leaders also indicated that they would like more control over development resources and time frames, but fewer indicated that they would improve these aspects of development.

Conclusions

Two aspects of the research results stand out. The first is that there are mechanisms which do significantly shorten software product development time. These include: sales and marketing involvement in defining product requirements, supplier involvement early and late in the development cycle, frequent load builds and system tests. The second is that low experience and high experience projects are different. The influence of the factors just listed is greater for low experience projects. In fact, there is little evidence for the influence of the commonly hypothesized determinants of development time for high experience projects.

The study should allow software development managers more effective use of the wider literature on the determinants of development time for greater competitive advantage.

References

1. Carmel, E., Cycle time in packaged software firms. *Journal of Product Innovation Management*, 1995, **12**, 110-123.
2. Datar, S., Jordan, C., Kekre, S., Rajiv, S. and Srinivasan, K., Advantages of time-based new product development in a fast-cycle industry. *Journal of Marketing*, 1997, **XXXIV**, February, 36-49.
3. Griffin A., The effect of project and process characteristics on product development time. *Journal of Marketing Research*, 1997, **XXXIV**, February, 24-35.
4. Stalk, Jr., G., Time: The next source of competitive advantage. *Harvard Business Review*, 1998, July-August,
5. Stalk, Jr., G. and Hout, T.M., *Competing Against Time*. The Free Press, New York, 1990.
6. Williams, J.R., How sustainable is your competitive advantage. *California Management Review*, 1992, **43**, 3,
7. Wheelwright, S.C. and Clark, K.B., *Revolutionizing Product Development*. The Free Press , New York, 1992.
8. Ittner, C. and Larcker, D., Product development cycle time and organizational performance. *Journal of Marketing*, 1997, **XXXIV**, February, 13-23.
9. Burchill, G. and Fine, C.H., Time versus market orientation in product concept development: empirically-based theory generation. *Management Science*, 1997, **43**, 465-478.

10. Zirger, B.J. and Hartley, J.L., The effects of acceleration techniques on product development time. *IEEE Transactions on Engineering Management*, 1996, **43,** 143-152.

11. Datar, S., Jordan, C., Kekre, S., Rajiv, S. and Srinivasan, K., New product development structures and time-to-market. *Management Science*, 1997, **43,** 452-464.

12. Eisenhardt, K.M. and Tabrizi, B.N., Accelerating adaptive processes: product innovation in the global computer industry. *Administrative Science Quarterly*, 1995, **40**, 84-110.

13. Hayes, R.H., Wheelwright, S.C., and Clark, K.B., *Dynamic Manufacturing*, The Free Press, New York, 1988.

14. Rosenau, M.D., Speeding your product to market. *Journal of Consumer Marketing*, 1988, **5**, 23-40.

15. Gupta, A.K., and Wilemon, D.L., Accelerating the development of technology-based new products. *California Management Review*, 1990, **32**, 2, 29-44.

16. Cordero, R., Managing for speed to avoid product obsolescence. *Journal of Product Innovation Management*, 1991, **8**, 283-294.

17. Ancona, D.G. and Caldwell, D.F., Bridging the Boundary: External activity and performance in organizational teams. *Administrative Science Quarterly*, 1992, **37**, 634-655.

18. Imai, K., Nonaka, I. and Takeuchi, H., Managing the new product development process: how Japanese companies learn and unlearn. In Hayes, R., Clark, K. and Lorenz, P. (eds.) *The Uneasy Alliance*. Harvard Business School Press, Boston, 1985, 337-375.

19. Clark, K.B. and Fujimoto, T., *Product Development Performance*. Harvard Business School Press, Boston, 1991.

20. Mabert, V.A., Muth, J.F. and Schmenner, R.W., Collapsing new product development times: six case studies. *Journal of Product Innovation Management*, 1992, **9**, 200-212.

21. Hartley, J.L., Meredith, J.R., McCutcheon, D. and Kamath, R.R., Suppliers' contribution to product development: an exploratory study. *IEEE Transactions on Engineering Management*, 1997, **44**, 258-267.

22. Jacobson, I., Christerson, M., Jonsson P. and Overgaard G.(1992), *Object-Oriented Software Engineering*. ACM Press/Addison Wesley, Reading, MA, 1992, 28.

23. Eisenhardt, K.M., Making fast strategic decisions in high velocity environments. *Academy of Management Journal*, 1989, **32**, 543-576.

24. Boehm, B. W., A spiral model of software development and enhancement. *IEEE Transactions on Computers*, 1988, **21**, 5, 61-72.

25. Cusumano, M. and Shelby, RW., *Microsoft Secrets*. The Free Press, New York, 1995.

26. Bowen, H.K., Clark, K.B. Holloway, C.A. and Wheelwright, S.C., *The Perpetual Enterprise Machine*. Oxford University Press, New York, 1994.