

Association for Information Systems AIS Electronic Library (AISeL)

ACIS 2004 Proceedings

Australasian (ACIS)

December 2004

Why Information Systems Development Projects are Always Late

Anthony Vlastic

Australian Graduate School of Management, UNSW

Philip Yetton

Australian Graduate School of Management, UNSW

Follow this and additional works at: <http://aisel.aisnet.org/acis2004>

Recommended Citation

Vlastic, Anthony and Yetton, Philip, "Why Information Systems Development Projects are Always Late" (2004). *ACIS 2004 Proceedings*. 120.

<http://aisel.aisnet.org/acis2004/120>

This material is brought to you by the Australasian (ACIS) at AIS Electronic Library (AISeL). It has been accepted for inclusion in ACIS 2004 Proceedings by an authorized administrator of AIS Electronic Library (AISeL). For more information, please contact elibrary@aisnet.org.

Why Information Systems Development Projects are Always Late

Dr Anthony Vlastic
Professor Philip Yetton
Australian Graduate School of Management

Australian Graduate School of Management
University of New South Wales
Sydney, NSW, Australia, 2052
Email: anthonyv@agsm.edu.au
Email: phily@agsm.edu.au

Abstract

Information systems development projects (ISD) are characterized by high rates of failure and escalation. To address this concern, in this paper a novel explanation for the poor performance is developed, which explores the influence two variables: task variance and task dependence. The findings suggest that the improvement in performance has been slow because of the interactive effect of the two variables driving both the level of, and variance in, performance. To address this problem, the paper concludes with a number of practical suggestions.

Keywords

Information systems development, project performance, variance

INTRODUCTION

The capacity to execute information systems development (ISD) projects on-time, in-budget, of-quality and to-scope is critical in today's economy, where a significant portion of new investment is driven by information technology and communications. Unfortunately, performance to date has been poor. For example, in a review of the empirical literature, Sauer (1999) found that from the late 1970s to today, failure rates have varied between 30% and 70%. This performance has remained low despite a tradition of research into ISD failure that dates back to the 1970s, and that has been in continuous development ever since (Sambamurthy & Kirsch, 2000).

Given this continued poor performance, the objective of this study is to examine an alternative, underdeveloped, viewpoint. The survey results, as outlined by Sauer (1999) and others, suggest one starting point. There are two critical performance characteristics reported in the survey results. One is the high expected rates of failure and overrun. For example, Johnson et al. (2001) report that 23% of projects are not complete and 49% of projects overrun their schedule and budget. Most ISD research has been concerned with reducing these rates of failures and/or magnitude of schedule overruns.

A close examination of the survey results reports in Sauer (1999) and others reveals a second characteristic, the large variance in the outcomes. That is, for projects that do not fail, but do overrun, the size of the overrun varies widely. Some variances are small, but others are large. Johnson et al. (2001), for example, report that schedule overruns vary from 0% to over 100%. Little prior research has explicitly focused on this schedule variance. An understanding of performance variance could, as it did in the study of production variances in total quality management (TQM) (Juran & Gryna, 1988), lead to overall improvements in ISD project performance. Drawing on the TQM literature, the premise here is that a reduction in task variance leads to both a reduction in schedule variance and a decrease in schedule overruns. The research question guiding this study is therefore:

How does task variance influence both the level of and variance in schedule overruns?

In addressing this question, the paper's contribution is twofold. First, it develops a framework to model the joint influence of task variance and task dependence on the project schedule. Task variance is a measure of the scatter or dispersion of the set of results, and task dependence is the degree to which the work performed on an individual task influences other tasks. Previous studies often estimate the strength of the relationship without demonstrating how or why it exists. Here, this limitation is overcome by considering both the factors that influence performance and the process by which those factors affect the schedule. Second, the study accounts for an unexplored component of the continued poor performance of ISD, the variance in schedule overruns. Previous studies have focused on the expected rates of overrun, with limited interest in the range of those overruns. Here, schedule variance is modeled as a critical dependent variable, rather than treated simply as a measure of residual error.

The balance of this paper is structured into four sections. First, the background literature and theoretical framework are outlined. Second, the methodology is described. Third, the framework is validated using literal replication across four case studies. Finally, the findings are discussed.

THEORETICAL FRAMEWORK

Research in ISD has traditionally been categorized as either factor or process based (Newman & Robey, 1992; Sambamurthy & Kirsch, 2000). A factor approach identifies potential predictor variables of successful performance (Newman & Robey, 1992). Examples include user involvement, management support and project planning (Yetton et al., 2000). In contrast, a process approach is where a project is defined as a sequence of tasks over time (Newman & Robey, 1992). Performance is explained by the relationships between those tasks. Examples include the learning, conflict, political and garbage can models (Newman & Noble, 1990). The factor approach is the dominant form used in ISD research (Sambamurthy & Kirsch, 2000).

Newman and Robey (1992) argue that insights from factor and process models can be complementary, with factor models establishing the relationship between predictors and performance, while process models examine the activities that generate those associations. Sabherwal and Robey (1995) show that although the ontological assumptions of factor and process models are different, their epistemological assumptions are the same. Therefore, they conclude that while the models consider different types of data, they have similar research goals.

Following this guidance, the framework developed in this paper considers both process and factor approaches. Specifying the process by which the factors influence performance significantly increases the explanatory power of the framework. To do this, a project is treated as an ordered sequence of tasks. The process description in this paper, which explains how tasks interact, is explained as two propositions and an assumption that model the critical development process. These clarify how two factors, task variance and task dependence, influence both the level of and variance in schedules. Before describing the process model, the definition of a project used in this paper is outlined in the next subsection.

Project Schedule

A project is a temporary undertaking to create a unique product (PMI, 2000). In order to manage a project from initiation through to closeout, a process needs to be followed to control resource allocation and task scheduling. This process establishes an ordered sequence for the tasks. Project scheduling is a technique used to allocate and monitor tasks in a sequence (Schwalbe, 2002). Gantt charts provide a standard format for displaying project schedule information as a bar chart that plots tasks on the vertical axis against time on the horizontal axis (PMI, 2000). Task duration is shown by date-placed horizontal bars. Figure 1 illustrates a Gantt chart with eight tasks. It provides a graphical illustration of the duration of tasks and their dependencies. From Figure 1, for example:

- Task C will take three months, and will begin when task B is complete
- Task G will take two and a half months, and start when both tasks E and F are complete.

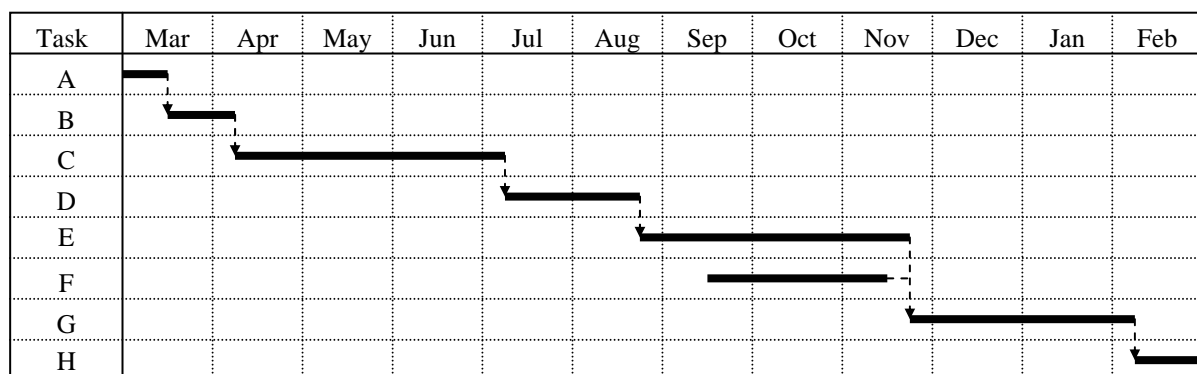


Figure 1: Gantt Chart

In developing a project schedule, a project manager considers the tasks to be completed, their duration and their dependencies. Project network techniques estimate the total project duration subject to those constraints. The techniques include Project Evaluation and Review Technique (PERT), Critical Path Method (CPM) and Graphical Evaluation and Review Technique (GERT), among others. PERT and CPM were developed in parallel, as separate approaches, in the late-1950s. They allowed project managers to identify the critical path for projects (Pich et al., 2002). These were the first techniques that moved scheduling away from a simple bar chart

and allowed for the analysis of multiple activities. With the introduction of GERT in the late-1960s, the ideas were extended to allow for probabilistic project outcomes and network loops (Clayton & Moore, 1972).

The 'critical path' on a Gantt chart is defined as the series of tasks that determine the earliest possible completion date for a project. For example, in Figure 1 the critical path is 12 months long, and the sequence is A-B-C-D-E-G-H. Task F is not on the critical path. The expected total project time is the sum of the time required for each task on the critical path. When tasks on the critical path overlap, the tasks are reclassified to remove the overlaps.

In practice, the time for each task varies within a range (task variance). So, in Figure 1, tasks on the critical path have multiple possible completion dates. This variation in completion dates for tasks leads to variation in completion dates for projects (project schedule variance). This can be illustrated with a Monte Carlo analysis, where, project time is modeled as a function of the tasks on the critical path (Schwalbe, 2002). While task variance has a direct effect on project performance variance as described here, its effect on the level of performance is more complex and little understood. The level of performance is considered in the next section.

Task Variance and Project Performance

Once a project schedule is developed, it is constrained by its dependencies. The Project Management Institute define three types of dependencies: mandatory, discretionary and external (PMI, 2000). Mandatory dependencies are present because one task relies on the completion of another task before it can begin. For example, a deliverable must be designed before it is developed. Further, a task usually depends on a number of predecessors. Thus, completing one task ahead of schedule may mean a wait before the next task begins. For example, in Figure 1 task G cannot begin until both tasks E and F are complete.

Discretionary dependencies are those at the judgment of the project team, such as allocating limited resources across tasks. The resources allocated to a project move from one task to another as required. Frequently, when a task is completed ahead of schedule, the next task has to be delayed until the required resources can be made available. For example, this could occur in Figure 1 if tasks E and F required the same resources, and so they cannot be completed in parallel as illustrated.

External dependencies are those that are beyond the control of the project team. For example, waiting for the delivery of a product from a vendor. In that case, when a project team completes a task early, they would have to wait for the next task. For example, assume in Figure 1 that task F is the installation of a vendor product. This cannot begin until the product is delivered to site. If the project team completes their tasks early, and is ready to do task F earlier than expected, it will not matter because they still have to wait for the delivery.

Each of these three dependencies limits the degree to which negative task variance can be used to offset positive schedule variance. In Figure 1, consider the hypothetical case where task E started one and a half months early. The flow on effect would be that tasks G and H could also start early. However, task G cannot start until both tasks E and F are complete, and task F completion date has not changed, so there would be very little variation in the start time for task G. When this occurs over a number of tasks, the dependencies decrease the likelihood that tasks on the critical path start early. The first proposition is therefore:

(P1) When task dependencies are high, tasks on the critical path are unlikely to start before their scheduled start time

One way of overcoming dependencies has been to introduce float/slack (Schwalbe, 2002). Project teams are experienced at managing small task overruns using tools such as float/slack. Such strategies however, have only limited applicability in environments where task variance is high. There are two reasons for this. First, in high dependency situations, only small levels of float/slack can be included before they have an impact on the critical path. For example, for task F in Figure 1, the maximum float/slack that can be included is one week, assuming task F cannot start earlier. Second, while actual task times frequently do not match the planned schedule, the issue is not whether this happens, but whether it was planned and allowed for in the project. Given that large variances are usually a disruption, the project team will avoid their occurrence, and so when they do occur they can be considered as not planned. Formally, the first assumption is thus:

(A1) Large positive task variances are unplanned

Combining P1 and A1, when task variances are large, and there are multiple task dependencies, the likelihood of a project team adjusting the schedule to accommodate them is low. This is because the overruns are unplanned (A1) and task dependencies restrict a project team's ability to adjust schedule when required (P1). Therefore, it follows that:

(P2) When both task dependence and task variances are high, project performance is both poor (relative to the sum of the estimated scheduled times for the tasks on the critical path) and unreliable (high variance in schedule overruns)

Proposition 2 is similar in form to the core proposition in TQM, in which there is *variance driven performance* (Juran & Gryna, 1988). Essentially, here the project critical path is analyzed as if it were equivalent to the manufacturing production line. TQM finds that increases (decreases) in task variance at any station on the line reduces (increases) production run times for the whole line, holding expected task cycle times constant. Estimating a project's critical path as the sum of the scheduled task times is subject to the same threat of variance driven performance. Of course, here, there is only one run down the production line. Current ISD methodologies focus on reducing the expected task cycle times and fast tracking projects, rather than reducing task variances to improve performance.

The influence of task variance and task dependence on performance variance and performance level for ISD projects is presented in Figure 2. The two factors, task variance and task dependence, interact to influence the performance level within a project and performance variance across projects. The methodology below describes how Assumption 1 and Propositions 1 and 2 are validated.

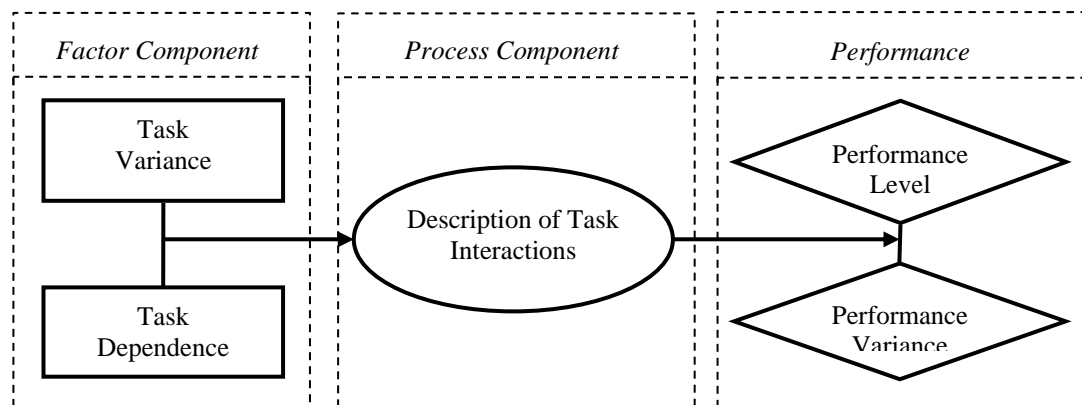


Figure 2: A Framework for Project Performance

METHODOLOGY

Four case studies are used to validate framework. The value of case study research is that it allows researchers to study IS in a natural setting for a previously little studied area (Benbasat et al., 1987). The cases are analyzed using a literal replication logic methodology designed to investigate whether similar results are repeated across different cases (Yin, 1984). The inclusion of each additional case strengthens the findings. Given the unit of analysis for this study is a project, the industry and organizational context are held constant. This increases confidence in the results. When projects are from different organizations, there would be a validity threat over whether the failure to replicate was a function of project, organisation or industry differences and, therefore, whether a project level analysis was warranted. So, the projects were selected from a single organization.

Organizational Context

The focal organization is a corporatized utility, WaterCo¹, responsible for the delivery of local water and wastewater services to 1.4 million people. Maintenance and operations are outsourced. Although a local geographical monopoly, the organization experiences considerable pressure from its customers, via the State government, to control cost and improve value for money.

The Information Systems (IS) Division reports to one of the six business units. IS work is typically outsourced, and the IS Division is described as a 'service broker'. The organization maintains only sufficient technical expertise to manage risk and draws on external consultants and vendors to provide technical skills and advice. These additional skills are usually obtained from one of the organization's strategic partners, with project management standardized on the PRINCE2 methodology.

Project Descriptions

Four projects were selected. To satisfy the literal replication logic, the projects had to be relatively recent or current, and have a significant influence on the operation of the business. To meet these criteria, the four projects chosen were large, had different goals, were located in different parts of the organization and had separate

¹ The organization, projects and individuals have been disguised for confidentiality.

management teams. The project characteristics are summarized in Table 1. Further details are available from the authors upon request.

The Variables

There are two key variables examined in this study, task variance and task dependence. Task variance is a measure of the scatter or dispersion of a set of scores. That is, the range of outcomes that can be realized on a task. For the case studies here, task variance is measured as the difference between the scheduled task completion times and the actual task completion times. In addition, this study is only concerned with large task variances. This is interpreted here to mean tasks that vary from their schedule by more than one week or 25% of the scheduled task cycle time, whichever is smaller. This is consistent with the control metrics used in WaterCo, namely, that differences from schedule of more than one week were reported at the weekly project meetings. It is assumed that the project team manages smaller variances, as part of their routine activities.

Task dependence refers to the degree to which the work performed on a task influences other tasks. In this study, the concern is the critical path for each project. By definition, the critical path involves high dependence. That is, each task is reliant on another task to be completed before it can begin.

Table 1: Project Characteristics

Project	SampProj	OpProj	LabProj	EquipProj
Project Description	Development of new system for gathering water samples.	Development of new work mgmt. system for pipe maintenance	Major upgrade of laboratory water quality testing system	Rollout of a new work mgmt. system for equipment maintenance
Client	Water Operations & Laboratory	Water Operations & Outsourced Water Operations Partner (WaterPartnerCo)	Laboratory	Equipment Maintenance
Project Cost	\$US1.0 million	\$US2.4 million	\$US0.35 million	\$US0.3 million
Project Duration^a	21 months	27 months	10 months	8 months
Project Methodology	PRINCE2	Consultant proprietary system	PRINCE2	PRINCE2

Notes:

- a) Project duration considers the time from the commencement of preliminary design through to closeout

Data Collection

Data collection included a triangulation of information from documentation, interviews and observations. All data was recorded and catalogued to make data gathering as explicit as possible. The primary source of information was documentation. This was used to establish the tasks that unfolded on the project and identify the size of any variance, where it occurred. The documents included manuals, procedures, plans, guidelines, project status reports, schedules, meeting agendas, meeting minutes, memos and lessons learned lists. This information covered the duration of each project, was well documented and was available on the organization's Intranet.

The documentation was complemented by interview and observations. The objective of interviews and observations was to establish leads that might be followed and confirm the account of tasks as described in the documentation. A total of 28 interviews were conducted of individuals across the projects. The observations were of meetings such as the Information Systems Executive Committee, chaired by the Chief Executive Officer, the Project Governance Office, chaired by the Chief Information Officer, and project teams.

Analysis

The analysis is designed to address the two propositions (P1 and P2), and the assumption (A1), by first exploring each project individually, and then comparing projects for similarities. The approach used to do this in this study was to characterize the events based on some external criteria. There are two reasons for adopting this approach. First, it removes some of the researcher bias in the analysis. Second, it provides a comprehensive list which limits errors of omission, overlooking a particular risk (Schmidt et al., 2001).

There were four stages to the analysis. First, a timeline was constructed for each case, identifying the scope, schedule and any changes over time, with a specific focus on the critical path. Second, a catalogue of tasks was developed for those tasks where the start or completion time diverted by more than one week, or more than 25%,

from the planned schedule. Third, each task variance was assigned to a category on the checklist. The checklist adopted for this study was developed by Schmidt et al. (2001) and included 14 groups². Finally, the tables were reordered so that the risks were grouped, and the variances associated with each risk listed. Comparisons of projects could then be made.

CASE STUDY RESULTS

Table 2 summarizes the findings. This section illuminates the merit of the two propositions, and one assumption, by considering characteristics of the tabulated results.

Proposition 1: Tasks Are Unlikely To Start Before Their Scheduled Start Time

The first proposition (P1) states that when task dependencies are high, tasks are unlikely to start before the scheduled start time. There are two conditions to consider for this to be the case. First, the dependencies need to be high. Given that in this study the critical path was being examined (as noted in ‘The Variables’ section above), by definition, the dependencies were high in each case. The second condition of P1 states that tasks are unlikely to start before their scheduled start time. This was also case on all four projects. There was no exception to this finding, namely, that a task starts more than one week earlier than it was initially scheduled. In some cases tasks on the critical path started a day or two early, however this did not affect the overall schedule. Thus, consistent with P1: when task dependence is high, tasks are unlikely to start before their scheduled start time.

Assumption 1: Large Positive Task Variances Are Unplanned

The framework developed and presented in Figure 2 assumes that large positive task variances are unplanned (A1). All four cases had tasks where the actual time varied by more than one week from the scheduled time (see Table 2). The issue here is whether they were, or could have been, planned when developing the initial schedule. In retrospect, it is easy to recognize potential problems. The question is ‘should’ they have been predicted. For example, on SampProj an on-going problem was the integration of three software products as part of the core system: geographic IS, reporting software and Microsoft .NET (*12.0 Technology*). At the time that the project plan was developed, the three vendors expected little difficulty in integration and were told that by the vendors.

Inspecting task variances reported for the four cases in Table 2, two insights are pertinent. One, it is not clear when a major overrun will occur. On initial inspection, it could appear that it should be predicted where there are similarities across the cases. However, while for example, three of the four projects had problems with the sign-off of documentation indicating sponsorship/ownership issues, this is a deviant instance. Instead, the overruns across the cases were different. For example, consider system stability at testing, which caused overruns in three of the projects (*6.0 Requirements*). Although this may seem like a similar issue for all projects, further investigation reveals that the reasons for the overruns are different. For SampProj, the problem was new software compatibility. For OpProj, it was new hardware/software compatibility. For LabProj, it was legacy system issues. Further, EquipProj, which also had a testing stage, did not have an overrun at this stage. Similarly, when the cases were faced with other similar problems, overruns did not occur in all cases. For example, SampProj did not have hardware/software compatibility issues, and OpProj did not have software compatibility problems.

Looking across the portfolio of issues for the projects, it is also clear in Table 2 that each project had its own unique set of problems. For example, SamProj had ten issues that caused a major overrun. Only three of these issues were repeated on OpProj and another seven emerged for that project. A similar comparison with other projects reveals that each has its own unique set of circumstances that are not repeated.

The other insight from Table 2 is that there is high variance in the magnitude of overruns. For example, on LabProj, there was a problem with the database conversion. Although it was expected that this might cause some concern, it was unclear what the extent might be, even though there was an initial risk assessment study. Similarly, in the case of SampProj, the data conversion task was partitioned out of the project, so that it would not cause delays. Another example is from OpProj, and the resolution of the WaterCo to WaterPartnerCo conflict. Although the project team recognized that this would take some time to resolve, it was unclear just how long the two parties would need to resolve the issue.

Thus overall, the case study findings are consistent with A1: large positive task variances are unplanned. Both the frequency of overruns and their magnitudes could not be predicted.

² This checklist was adopted because, as Schmidt et al. (2001) note, it is currently the most authoritative checklist available. In the Schmidt et al. checklist, the 14 groups represent 53 items. These items account for 29 of the 33 items from a combined checklist of previous studies. In addition, Schmidt et al. identify 26 new items.

Proposition 2: Poor and Unreliable Performance

Proposition 2 (P2) states that when both task dependencies and task variances are high, project performance is both poor (relative to the sum of the estimated schedule times for the tasks on the critical path) and unreliable (high variance in schedule overruns). The cases support the proposition. The overruns reported on the four projects are 25 weeks (SampProj), 12 weeks (OpProj), 23 weeks (LabProj) and 7 weeks (EquipProj). All were late, with performance was poor relative to the planned schedule. In addition, the magnitudes of the overruns are different, and are different as a percentage of the total planned schedule. The percentages are 76% (SampProj), 26% (OpProj), 105% (LabProj) and 33% (EquipProj). Project performance against schedule was unreliable.

DISCUSSION AND IMPLICATIONS

Managing ISD Projects

The concern for this paper was to explain the continued poor performance of ISD projects and in particular the influence of task variance. One reason for the poor performance, which is commonly mentioned in the press and other literature, is that it is the result of poor planning. The framework presented in this paper suggests a different viewpoint. It suggests that poor performance has two components: performance level and performance variance. Further, the critical characteristics to consider are:

- High task dependence, restricting task start times
- Unplanned high positive task variance
- Poor and unreliable performance as a result of high task dependence and unplanned task variance.

In considering these characteristics, it is important to recognize that task variance and task dependence are not two variables operating in isolation. It is their combined influence that is the most significant. If a task's completion time varies from the plan, this would have repercussion only where it affects the critical path, through the dependencies. Otherwise it has little influence. For example, on SampProj, data conversion was separated out from the core project at an early stage, so the variance in these task outcomes would not affect the rest of the project. In contrast, there were a number of system dependencies for SampProj causing delays.

Also, the dependencies are not a concern unless a task does not meet its planned completion time. Tasks that are on time or are completed early satisfy the schedule requirements, and so do not change the schedule. It is also worth noting the influence of small variances. Although many tasks do not fulfill their initial requirements, problems only arise where there is a large overrun. In general, small overruns could be accounted for by minor variations in the project schedule.

In addition, it is easy to suggest that the problems that occurred on the projects should have been predicted, and that better performing projects are just better managed. Although there is some truth in that, it is not clear from the cases. All project had strong teams managing them, they all just had problems in different areas.

Thus, consistent with the framework (Figure 2), it is the combined influence of task variance and task dependence that has the most significance on project performance. Task dependence restricts the start time of tasks, and large task variances, where they occur, are a random outcome in terms of the size of overrun. In the next subsection, the implications of this for organizational learning are considered.

Learning

Over time, it would be expected that organizations learn, and so improve performance. This has not happened though in practice. If the survey of Sauer (1999) and Johnson et al. (2001) results are any guide, performance problems persist. An implication of the framework is that it explains why this is the case. That is, it explains why project performance is poor, and why learning is difficult. Consider the passage that follows to illustrate.

A fast-tracked project, which includes a number of tasks with high performance variance, incurs a major overrun. This outcome would be in the expected range, based on the Figure 2. That is, it is only one possible result from a large random set. Although it should be expected, the overrun would typically be interpreted as the result of inadequate project management. Actions would be put in place in the next project to improve performance on tasks similar to those with high overrun. Of course, the action would be 'successful' the next time. A similar type of mistake would not be repeated. However, if task variance remains high, other tasks would exhibit significant overruns. An example from the cases is that three of the projects had problems while testing, but all the problems were different. The risk has been identified generally, but the specific issues for that risk is a project-by-project proposition.

Thus, under these conditions, learning would be slow, in general, because attempts to learn would often be in response to errors or 'noise' in project performance. That is, large positive task variances are unplanned. As a result, a fundamental cause of poor performance, the process loss due to high task variance combined with high task dependence, has not been identified and addressed. This interpretation is supported by the feedback literature. In a meta-analysis of the use of feedback interventions, Kluger and De Nisi (1996) found that not all feedback is valuable, and at least one third actually decreases performance. This is also consistent with the core proposition of TQM (Juran & Gryna, 1988). That is, major improvement in performance can be gained by reducing the variance on events that make up a whole.

Initiatives to Improve Project Performance

This study suggests that to improve performance there is two criteria that should be examined: task variance and task dependence. If either of these can be reduced this will improve performance. Table 3 suggests some alternatives, with examples from the cases to illustrate. The thrust of the suggestions is on the appropriate framing of a project so that many of these difficulties do not have to be faced. None of the ideas presented are new. For example, some of the more modern, agile, methodologies already have implicit in them some of the principles listed in the Table 3 (see for example McConnell (1996)). The value of the table is that it makes explicit the suggestions, and consolidates ideas that have been proposed in various different areas.

Table 3: Initiatives to Improve Project Performance

Variable	Initiative	Explanation	Example From Case Studies
Task Variance	Increase task transparency	Allows for better planning in other areas	<i>Organization:</i> PRINCE2 adopted as standard.
	Increase user participation	Ensures that products delivered meet user needs	<i>OpProj:</i> Conducted several workshops
	Manage expectations	Lower expectations have a higher chance of success	<i>SampProj:</i> Scope scaled back to only mandatory requirements
	Use packaged software	Provides a standard to develop the system	<i>OpProj:</i> Choice of vendor package that would need little customization
Task Dependence	Scrub requirements	Removing tasks will reduce the number of dependencies	<i>SampProj:</i> Scope revised to accommodate development in a related project
	Improve requirements definition	Ensures that there is no confusion over what is to be developed and when	<i>LabProj:</i> A full specification of what was thought to be in use was sent to the key users for confirmation before start
	Reduce task coupling	If task links are reduced then dependencies exert less influence	<i>EquipProj:</i> Each workshop implementation was undertaken as a sub-project
	Stage projects	Reduces the direct dependencies	<i>LabProj:</i> Emergent schedule, with three stages

Limitations

There are several limitations to this study. Three of the more significant are mentioned here. First, one organization was selected, which limits generalizability. While this is unusual for multiple case study research, it was integral to the research design, because of the project level of analysis. In addition, the structure of the organization is not unusual, in that there are several business units, and these are serviced by a separate IS Division. However, the findings should be generalized only to similarly structured organizations. Future research should consider other organizational forms, and other industries.

Second, much of the study data was based on retrospective documented accounts. This could result in some distortion of the facts. However, this study triangulated data across several sources including interview, observations and documents. This increases confidence in the findings. Future research might consider a longitudinal study.

Finally, in analyzing the data the task variances were characterized into groups. This classification could reflect reviewer interpretation, rather than actual. Although all care was taken to ensure that the categories did reflect the actual tasks, some concern remains. Future research could consider other frameworks to confirm results.

SUMMARY

The objective of this paper was to determine how task variance influences the level of and variance in performance. Factor and process approaches were combined to undertake the study. Having considered four case studies, the two propositions and one assumption are supported. The framework suggests that the interaction between the two variables - task variance and task dependence - has a significant influence on the project outcome, through the development of the project schedule. Such a framework provides an alternative way of thinking about the performance improvement process. Research to date has tended to focus on the level of estimates, with little regard to the variance. For firms to significantly improve the performance of their projects, they should focus on activities that address both of the key variables. Table 3 provides some recommendations.

REFERENCES

- Benbasat, I., Goldstein, D. K., & Mead, M. (1987) The Case Research Strategy in Studies of Information Systems. *MIS Quarterly*, 11(3), 369-386.
- Clayton, E. R., & Moore, L. J. (1972) PERT vs GERT. *Journal of Systems Management*, February, 11-19.
- Johnson, J., Boucher, K. D., Connors, K., & Robinson, J. (2001) Project Management: The Criteria for Success. *Software Magazine*, 21(1), S3-S11.
- Juran, J. M., & Gryna, F. M. (1988) *Juran's Quality Control Handbook*. New York, USA: McGraw-Hill.
- Kluger, A. N., & De Nisi, A. (1996) The Effects of Feedback Interventions on Performance: A Historical Review, a Meta-Analysis and a Preliminary Feedback Intervention Theory. *Psychological Bulletin*, 119(2), 254-284.
- McConnell, S. (1996) *Rapid Development: Taming Wild Software Schedules*. Redmond, USA: Microsoft Press.
- Newman, M., & Noble, F. (1990) User Involvement as an Interaction Process: A Case Study. *Information Systems Research*, 1(1), 89-113.
- Newman, M., & Robey, D. A. (1992) A Social Process Model of User-Analyst Relationships. *MIS Quarterly*, 16(2), 249-266.
- Pich, M., T., Loch, C. H., & De Meyer, A. (2002) On Uncertainty, Ambiguity and Complexity in Project Management. *Management Science*, 48(8), 1008-1023.
- PMI. (2000) *A Guide to the Project Management Body of Knowledge*. Pennsylvania, USA: Project Management Institute.
- Sabherwal, R., & Robey, D. (1995) Reconciling Variance and Process Strategies for Studying Information System Development. *Information Systems Research*, 6(4), 303-327.
- Sambamurthy, V., & Kirsch, L. J. (2000) An Integrative Framework of the Information Systems Development Process. *Decision Sciences*, 31(2), 391-411.
- Sauer, C. (1999) Deciding the Future for IS Failure: Not the Choice You Might Think. In W. L. Currie & R. Galliers (Eds.), *Rethinking Management Information Systems: An Interdisciplinary Perspective*. Oxford, UK: Oxford University Press.
- Schmidt, R., Lyytinen, K., Keil, M., & Cule, P. (2001) Identifying Software Project Risks: An International Delphi Study. *Journal of Management Information Systems*, 17(4), 5-36.
- Schwalbe, K. (2002). *Information Technology Project Management*. Cambridge, MA: Thomson Learning.
- Yetton, P., Martin, A., Sharma, R., & Johnston, K. (2000) A Model of Information Systems Development Project Performance. *Information Systems Journal*, 10(4), 263-289.
- Yin, R. K. (1984) *Case Study Research: Design and Methods*. Beverly Hills, CA: Sage Publishing.

COPYRIGHT

Anthony Vlastic and Philip Yetton © 2004. The authors assign to ACIS and educational and non-profit institutions a non-exclusive license to use this document for personal use and in courses of instruction provided that the article is used in full and this copyright statement is reproduced. The authors also grant a non-exclusive license to ACIS to publish this document in full in the Conference Papers and Proceedings. Those documents may be published on the World Wide Web, CD-ROM, in printed form, and on mirror sites on the World Wide Web. Any other usage is prohibited without the express permission of the authors.