8-25-1995

The Effects of Task Interdependence, Goal Conflict, and Coordination Strategy on Software Project Success

Hayward P. Andres
Portland State University

The Effects of Task Interdependence, Goal Conflict, and Coordination Strategy on Software Project Success

Hayward P. Andres
School of Business Administration
Portland State University

1. Introduction

Recent research on managing software development has suggested that managing software development projects requires an ability to understand and balance the technological (e.g. tools, methodologies), economic (e.g. cost and effort estimations), and social bases (e.g. coordination strategies, goal structures, organizational climate) through which large software systems are developed (Boehm & Ross, 1989; Scacchi, 1984). Researchers have also noted that, whether a project structure is characterized by functional teams (i.e. design team, programmer team, and user group) or cross-functional teams (i.e. multiple teams, each comprised of designers, programmers, and users), project management's primary problems are the coordination of efforts and conflict resolution (Curtis et. al, 1988). The research questions addressed in this study demonstrate an inquiry into whether task interdependence and goal conflict each interact with managerial coordination strategies to influence software project success. Interactions among coordination strategies, task interdependence, and goal conflict will also be addressed. The choice of a specific coordination strategy is dependent upon managerial assessments of structural and task characteristics of the organizational context that are perceived to be affecting software development performance (Mantei, 1981; Henderson & Lee, 1992; Kim & Umanath, 1993). The model for this research is shown in Figure 1 below.

2. Project Success

Project success has been defined as a combination of two types of implementation outcomes: task outcomes and perceived team psychosocial outcomes (Pinto & Pinto, 1990). Software project task outcomes typically refer to adherence to the estimated schedule and budget and optimal productivity in terms of delivered source code instructions per man-hour (Henderson & Lee, 1992). Dimensions of software quality such as user-friendliness and adherence to end-user specifications are also associated with software development task outcomes (Yeh, 1993). Team psychosocial outcomes refer to the evaluation of the degree of experienced friendliness and support, positive feelings associated with interactions, acquired knowledge and skills, enjoyment of participation, and sense of pride and value resulting from participation in project implementation (Pinto & Pinto, 1990).

3. Coordination Strategy

Coordination refers to the integrating or linking together of different parts of an organization to accomplish a collective set of tasks (Van De Ven et. al., 1976). Informal,

cooperative, and decentralized strategies are characterized as organic strategies, while formal, controlling, and centralized strategies represent mechanistic strategies (Burns & Stalker, 1961). Research has found that the degree of task interdependence (Van de Ven et. al., 1976; Victor, 1990), degree of differentiation (Lawrence & Lorsch, 1967; Victor, 1990), and degree of goal conflict (Victor, 1990) are all related to selecting between organic and mechanistic coordination strategies.

4. Task Interdependence, Coordination Strategy, and Project Success

Task interdependence refers to the extent to which a task requires organizational units to engage in workflow exchanges of products, information, and/or resources and where actions taken in one unit affect the actions and work outcomes of another unit (Thompson, 1967). Pooled (or low) interdependence is evidenced in the software development context by the development of standalone segments of a software system where systems integration is an additive process (Kim & Umanath, 1993). According to Mantei (1981) these standalone segments are high modularity programming tasks that can be completely compartmentalized or split into subtasks where minimal intragroup and/or intergroup communication for task execution is needed. Alternatively, strategic systems applications often involve selective integration of decision support tools (e.g., expert systems, analytical packages, neural networks) with conventional transaction processing (Kim & Umanath, 1993). The developmental activities of such applications are typically delegated across diverse technically skilled development personnel or multiple cross-functional teams that must work in concert to complete the tasks. According to Kim and Umanath (1993), the development of such strategic systems applications is characterized by reciprocal workflow or reciprocal (or high) interdependence.

Figure is not available. Please contact author.

Thompson (1967) noted that as task interdependence increases, it is predicted that managers would select increasingly local, informal, and cooperative (organic) coordination strategies in order to reduce coordination costs. Van de Ven, et. al. (1976) found that increasing amounts of interdependence were positively associated with the use of lateral coordination strategies such as horizontal communication channels and unscheduled meetings (i.e. organic strategy).

In high interdependent software development tasks, interfaces (i.e. file structures and module input/output parameters) among interdependent program modules, developed by distinct software subteams, require iterative refinement through collaboration and negotiation among software project subteams. Increasingly local, informal, and cooperative (organic) coordination strategies effectively facilitate the needed lateral communication and interpersonal and/or subteam interactions. In contrast, centralized, formal, and controlling (mechanistic) coordination strategies are cost effective where low interdependence between program modules reduces the need for lateral interactions among the subteams. This suggests the following hypothesis with respect to software development:

H1: Software projects characterized by greater (less) task interdependence will be more successful given the use of an organic (mechanistic) coordination strategy.

5. Goal Conflict, Coordination Strategy, and Project Success

Boehm and Ross (1989) noted that: 1) end-users desire user-friendly systems with adequate functional task support; 2) customers desire delivery of a product of high quality, reliability, on short schedule and low budget; and 3) designers desire interesting technical challenges and fast career paths. These desires are instances of goal conflict with respect to the software development product and process outcome. Robey, et. al. (1989) observed that participation and influence in decision-making during systems development are essential elements of conflict generation, and ultimately, opportunities for conflict resolution.

When competitively linked goals lead to overt behavior among the software subteams, mediated confrontation using centralized, formal, and controlling (mechanistic) coordination strategies are required to resolve conflict and maintain superordinate goals (Victor, 1990). Local, informal, and cooperative (organic) coordination strategies are more suitable to situations of cooperatively linked goals where overt behavior is at a minimum (Victor, 1990). This assessment of past research on goal conflict and coordination strategies suggests the following hypothesis:

H2: Software projects characterized by greater (less) goal conflict will be more successful given the use of a mechanistic (organic) coordination strategy.

6. Task Interdependence, Goal Conflict, Coordination Strategy, and Project Success

The coordination problems associated with the degree of task interdependence cannot be conceptualized as independent of the degree of goal conflict (Victor & Blackburn, 1987). A state of increasing task interdependence gives rise to increasing potential for perceived opportunity for interference with one another's goal attainment. For example, Mitchell and Silver (1990) reported that in a tower building task in which subjects worked under conditions of task interdependence, a group goal prompted cooperative strategies, and these strategies resulted in greater cooperation and higher performance. In contrast, individual goals led to competitive strategies that focused on maximizing individual performance at the expense of group performance.

Under *norms of rationality*, it would appear that software project managers would organize (i.e. more assertive hierarchical or forcing methods) to handle real conflict resulting from incompatible goals among software subteams in precedent over problems arising from task interdependence attributed to program module assignment. In other words, when the degree of goal conflict is high, increasing task interdependence will lead to the selection of increasingly mechanistic strategies (Victor, 1990). High task interdependence typically warrants the use of organic coordination strategies. Thus, interactions between task interdependence, goal conflict, and coordination strategies are predicted. It is hypothesized that:

H3: Software projects characterized by greater task interdependence and greater goal conflict will be more successful given the use of a mechanistic coordination strategy.

7. Research Design

In this study, subjects were drawn from a population of computer science and management information systems undergraduate and graduate students familiar with the C programming language. A microcomputer platform utilizing the Turbo C programming language running under the MS-DOS operating systems was selected to implement the coding and testing phases of the program development task. The design of the experiment was a 23 factorial design. Factor A was task interdependence, high or low; factor B was goal conflict, correspondent or noncorrespondent; and factor C was coordination strategy, mechanistic or organic. The dependent variable is a multiple measure of project success which is comprised of task outcomes (i.e. schedule adherence, productivity, product quality) and team psychosocial outcomes (i.e. interaction quality, solution satisfaction, and process satisfaction).

Multiple regression analysis was used to conduct factorial ANOVA to assess the main effects and interactions of the research factors. For each dependent variable, an ANCOVA was performed in order to test the significance of the covariates. In addition, the ANOVA model was appropriate in assessing the interaction effects between the independent variables.

8. Summary

Tentative findings suggests a negative impact on productivity and negative affective responses in increased interdependence accompanied with the use of a mechanistic coordination strategy. This research utilizes a controlled experiment in order to investigate factors associated with the social context of software development that impact software development productivity. Laboratory-based experimentation affords programmatic research which facilitates learning (Dickson, 1989). Dickson (1989) suggested that experimentation helps to substantiate concepts and focuses on theoretical development aimed at understanding why a phenomena occurs. In addition, experimentation affords researchers with greater control than other methods and accessible experimental subjects with attributes close to that of their real world counterparts. This research should contribute to improved understanding of the process dynamics of workgroups engaged in software development. The understanding of the dynamics of software development workgroups could enable project managers to develop appropriate coordination strategies that adequately address human and structural factors that may contribute to software backlogs and high costs.