- **Increased Phase 2 coordination demand:** Discovering released Phase 1 errors generates demand for coordination activity in Phase 2.

- **Reduced labor available for Phase 2 basework and quality assurance:** Increased Phase 1 rework and coordination activities reduce the labor available for basework and quality assurance. This further slows the completion of Phase 2.

The numerous impacts of error generation also help explain why the Basic Probability of Flawed Task is so influential on model behavior.

5.5.4 Model Calibration Summary

The Python development project conditions are consistent with the Product Development Project Model boundary assumptions. Parameter values based on field data from the Python development project were used to calibrate the model in single phase and multiple phase configurations. The model simulated the fundamental behavioral characteristics of several project performance data sets reasonably well. Causal loop diagrams were used to explain how model structure causes the model behavior. Generally, available work and resource quantity structures were found to impact behavior more than resource effectiveness and project target structures. The ability of the model to describe the Python project's behavior increases the confidence in the model's use for policy analysis. The next section applies the model to policy analysis within the context of projects like the Python development project.

5.6 Policy Analysis

This section demonstrates the use of the Product Development Project Model for analyzing a project management policy. This will be done by investigating the impacts of a single type of management policy, coordination, on project performance. Despite being widely discussed in the business and research literature as a means of improving project performance coordination is not as precisely defined as some policies (e.g. a tax rate) or as well understood. Therefore a brief review of the project coordination literature sets a context for the analysis. This is followed by a description of how coordination policies can be represented in the Product Development Project Model and the descriptions of three coordination policies. The impacts of those policies on project performance in a project similar to the Python development project will be the basis for a deeper analysis of the effects of specific coordination parameters on performance. The model structure is used to explain the behavior and expand the understanding of coordination. The potential impacts of the analysis for coordination policy design and summaries complete the chapter.

5.6.1 Literature Review of Product Development Project Coordination

5.6.1.1 Traditional Characterizations of Coordination of Product Development Projects

Traditional methods of coordination are based upon a functional sequential product development paradigm for project structure (Watton, 1969; Wheelwright and Clark, 1992; Hayes et al., 1988; Clark and Fujimoto, 1991; Zaccai, 1991; Nevins and Whitney, 1989). The manufacturing development organization associated with the traditional development paradigm is based on separate functional units, typically marketing, engineering or R & D, and manufacturing. Coordination is seen as necessary primarily within the organization. Therefore organizational design and internal coordination mechanisms are the primary methods of coordination within the traditional development paradigm.

While often described as a structure for an entire organization (e.g., Wheelwright and Clark, 1991), the matrix structure can also be applied to individual development projects. Boeing's organization for developing the 727-100 is an example of a matrix structure for a single development project (Maxam, 1978). Technical Staff Engineers represented functional departments. These Technical Staff engineers worked closely with Project Design Engineers. Project Design Engineers represented portions of the aircraft in a manner analogous to project managers representing different development projects. The integration of the functional and project aspects of the aircraft through these two formal leaders was facilitated by engineering specialists "who each seemed to be a favorable blend of Project Design and Technology Staff experience."

When project managers act as coordinators in the traditional development paradigm, such as in a matrix structure, their role and effectiveness are limited. The role is limited by the perception of coordination as an activity to be performed primarily within the organization. The project manager's effectiveness is limited by their relatively low level of authority when compared to functional department heads. The limitations which low levels of authority place on project managers as coordinators and their effects are illustrated by Clark and Fujimoto (1991b) and

Womack et al. (1990). These researchers describe project managers in the traditional paradigm as part of the American and European automobile industry in the 1980s. In one case Womack et al. (1990) describe the contributions of this approach to project management and poor coordination to the two year (40%) delay in the development of General Motors's GM-10 model.

5.6.1.2 Recent Characterizations of Coordination of Product Development Projects

The more recent concurrent cross-functional development paradigm responds in at least two ways to the increased coordination needs caused by increased dynamic environments and interdependencies: expand the scope of coordination and integrate with cross-functional development teams.

The need for increased coordination between development organizations and their environments has been articulated by several researchers. Many participants which are not a part of the immediate project team or product development organization have been identified as requiring coordination, including customers (Bacon, 1994; Fujimoto et al., 1992; Ulrich and Eppinger, 1994; Wheelwright and Clark, 1992 Clark and Fujimoto, 1991b; Wheelwright and Sasser, 1991; Hauser and Clausing, 1988), technology (Iansiti, 1993b, 1992; Iansiti and Clark, 1993; Hayes et al., 1988), the distribution chain (Wheelwright and Clark, 1992), competitors (Bacon, 1994; Wheelwright and Clark, 1992; Hauser and Clausing, 1988), and regulation and standards (Bacon, 1994; Vogel, 1993).

Cross-functional development teams are groups of development specialists from different functional domains who work together on a single development project. Construction projects have utilized cross-functional development teams for many years. The use of cross-functional teams in manufacturing firms has been closely linked to product development performance (Cooper and Kleinschmidt, 1994). As a tool for coordination the formation of cross-functional development teams is an extension of the move away from functional-based groups to the matrix structures used in the traditional development paradigm. Hayes et al. (1988) describe and Wheelwright and Clark (1992) later refine a more detailed model of this shift with intermediate steps defined by the level of influence of project managers.

Boeing's 777 project provides an example of using cross-functional development teams for coordination (Peterson and Sutcliffe, 1992; Stix, 1991). Boeing modified the matrix structure used to develop its 727-100 for the development of the 777 aircraft. Chief Engineers led functional domains such as propulsion, avionics, structures, electrical, the flight deck, and aerodynamics. They were responsible for functionality, reliability, maintainability, manufacturability, cost, and certification. Chief Project Engineers were positioned orthogonal to the Chief Engineers in the matrix structure. They were each responsible for at least one of the airplane's sixty-five individual systems. Additional Chief Project Engineers integrated these individual systems within the airplane as a whole and integrated the development project with external participants such as customers and certification testing. Boeing formed over 270 cross-functional development teams within this structure. Peterson and Sutcliffe's (1992) description of the teams illustrates the cross-functional nature of their role: "These teams are defined around individual airplane systems, and are working cross-systems integration and vertical development issues (life cycle) simultaneously."

5.6.1.3 Coordination Tools, Expected Influences and Benefits

According to the literature product development project coordination can take several forms, including:

- The use of cross-functional development teams with time-dedicated (to the project) members who have balanced stakes in project success and commitment to project success (Cooper and Kleinschmidt, 1994; Rosenthal, 1992; Nevens et al., 1992; Dean and Susman, 1992; Clark and Fujimoto, 1991)

- Strong leadership or project champions (Cooper and Kleinschmidt, 1994; McCord and Eppinger, 1993; Clark and Fujimoto, 1992, 1991)

- Documentation which more clearly defines the development space for all developers. Examples include network diagrams, Quality Function Deployment development checklists or "Lessons Learned" books (Rosenau and Moran, 1993; Bacon et al., 1994; Nevens et al., 1992; Dean and Susman, 1992; Wheelwright and Sasser, 1992; Suh, 1990; Hauser and Clausing, 1988).

- Tools for sharing cross functional knowledge such as collocation and advanced technologies (Murmann, 1994; McCord and Eppinger, 1993; Morelli and Eppinger, 1993; Rosenthal, 1992; Peterson and Sutcliffe, 1992; Nevens et al., 1992; Wheelwright and Clark, 1992; Clark an Fujimoto, 1991)

- Specific positions, teams, or departments dedicated to project coordination (McCord and Eppinger, 1993; Peterson and Sutcliffe, 1992; Nevens et al., 1992; Dean and Susman, 1992; Clark and Fujimoto, 1992)

- Integrating organization structures such as matrix versus functional structures and integrating processes such as systems engineering (Ward et al., 1995; Bacon et al. 1994; McCord and Eppinger, 1993; Peterson and Sutcliffe, 1992)

The literature indicates that these coordination efforts will result in the following changes in a project:

- Earlier agreement on a sharp product definition (Cooper and Kleinschmidt, 1994; Murmann, 1994; Clark and Fujimoto, 1992)

- Increased capability for effective concurrent development, meaning more parallelism in activities will reduce risks and cycle time without degradation of other performance measures (Murmann, 1994; Clark and Fujimoto, 1991)

- Fewer and faster development iterations (Murmann, 1994; Gomory, 1992; Nevens et al., 1992; Wheelwright and Clark, 1992; Clark and Fujimoto, 1991)

- Improved communication in quality, quantity, and timing leading to shared mental models (Clark and Fujimoto, 1991; Rosenthal, 1992; Wheelwright and Clark, 1992; Wheelwright and Sasser, 1992)

- Increased commitment, trust and joint responsibility in the development team (Wheelwright and Clark, 1992; Clark and Fujimoto, 1991)

According to the literature these product development project coordination changes can improve all three primary measures of project performance:

- Reduced cycle time (Cooper and Kleinschmidt, 1994; Iansiti and Clark, 1993; Rosenthal, 1992; Nevens et al., 1992; Wheelwright and Clark, 1992; Clark and Fujimoto, 1991)

- Improved product quality (Iansiti, 1993; Rosenthal, 1992; Nevens et al., 1992; Wheelwright and Clark, 1992; Clark and Fujimoto, 1991)

- Reduced cost (Iansiti, 1993; Nevens et al., 1992; Wheelwright and Clark, 1992; Clark and Fujimoto, 1991)

The case in the literature for improving project performance by increasing coordination is compelling. However the record of attempts to apply coordination policies for improved performance is decidedly mixed. Wheelwright and Clark (1992) cite a case in which unsuccessful cross-functional teams *increased* cycle times. Reasons for coordination failures cited by other researchers vary. Dean and Susman (1991) found friction between members of the team from design and manufacturing. Wheelwright and Sasser (1991) cite a lack of planning due to a lack of information. Nevens et al. (1991) identified a lack of cross-functional skills in team members and no one taking responsibility for coordination. Clark and Fujimoto (1991b) found an automobile development team consisting of only liaison people and no developers. The team failed because it was ignored by those developing the product. Contributing to the lack of understanding is the apparently successful coordination of product development projects by other firms without using several of the most commonly cited coordination tools such as collocated dedicated cross-functional teams and frequent meetings (Ward et al., 1995). Some research points to different impacts of coordination of different measures of project performance. Iansiti (1993) found that increased coordination of product development in the automobile industry was related to improvements in quality, cycle time and productivity but not to quality alone. The experience of the Bose Corporation described in the first chapter is an example in which re-engineering the product development process for improved coordination increased one performance measure (quality) but not another (cycle time). Several researchers have identified the need to understand the causal relationships which link coordination to project performance (Adler et al., 1995; Cusumano and Nobeoka; 1991).

Why do some increased coordination policies produce significantly improved project performance while others do not? What impacts do increased coordination have on the internal workings of a project that impact performance?

5.6.2 Model Representation of Coordination Policies

Coordination policies can be investigated with the Product Development Project Model by describing different coordination policies with model parameter values and analyzing the resulting project performance through the model structure.

5.6.2.1 Coordination Parameters

Coordination policies are represented in the Product Development Project Model with four parameters:

- **The Coordination Labor Delay** is the gap between the demand for coordination and the application of labor to coordination. It represents the response time of the development team to the need for coordination. This response time would be expected to decrease due to the use of coordination tools such as cross-functional teams, collocation, integrating processes such as regular coordination meetings, and advanced communication technologies.

- **Coordination Priority** is the importance given to the coordination activity relative to basework, quality assurance, and rework. This parameter would be expected to increase with the use of strong project managers, product champions, special coordinating entities and integrating structures and processes.

- **The Coordination Minimum Task Duration** is the average time required to perform a coordination task by a developer within a specific phase. This parameter includes coordination meeting times and integration team activities. This parameter is expected to increase to account for the additional time invested in coordination.

- **The Time to Adjust Coordination Productivity Expectations** describes how quickly developers report the productivity of their coordination efforts and translate them into expectations about future coordination productivity. This parameter would be expected to decrease with increased awareness and focus on coordination.

The literature indicates that adjusting the Product Development Project Model coordination parameters in the directions described above will produce improved project performance in two (cycle time and cost) or all three performance measures.

5.6.2.2 Three Coordination Policies

Three coordination policies will be used for analysis:

- **No Coordination:** This policy prevents coordination activities, allowing the backlog of tasks needing coordination to build up. Although rare or nonexistent in practice, this policy establishes a baseline for comparison of other policy impacts and analysis.

- **Passive Coordination:** This policy reflects a perspective that coordination is a "necessary evil" in development projects. Coordination is left primarily up to individual developers to perform and manage as they see fit. This policy permits coordination work in response to coordination needs but gives it lower priority than basework, quality assurance, or rework and allows significant delays in coordination aspects of the project. This policy closely resembles the Python project in which coordination consisted primarily of the use of a cross-functional development team.

- **Active Coordination:** This policy reflects a perspective that proactive coordination is essential to project success. It considers coordination to be as important as other project development activities. This policy implements coordination tools and otherwise works to reduce delays in performing coordination and communicating about coordination. As a result more time is spent coordinating tasks needing coordination.

The values for the coordination parameters used to describe the three coordination policies are shown in Table 5-4. They were applied to all four development phases equally.

	Coordination Policy		
	Passive Coordination	*No Coordination*	*Active Coordination*
Coordination Parameter			
Labor Delay	NA*	6	2
Priority	0	1	4

Minimum Tasks Duration	NA*	1.0	1.25
Adjust Prod Expect. Time	NA*	1.5	0.5

* - Not Applicable. The No Coordination policy precludes any coordination labor. Therefore the Coordination Labor Delay and Coordination Minimum Task Duration parameters have no impact.

Table 5- 4: Model Parameter Values for Three Coordination Policies

5.6.3 Project Performance Under Different Coordination Policies

Figures 5-28, 5-29, and 5-30 show the Task Released phase behavior of a project closely resembling the Python Project using the three coordination policies. The relatively early (week 65) first release of design tasks and no rework of returned flawed tasks is evident in the no coordination policy simulation.

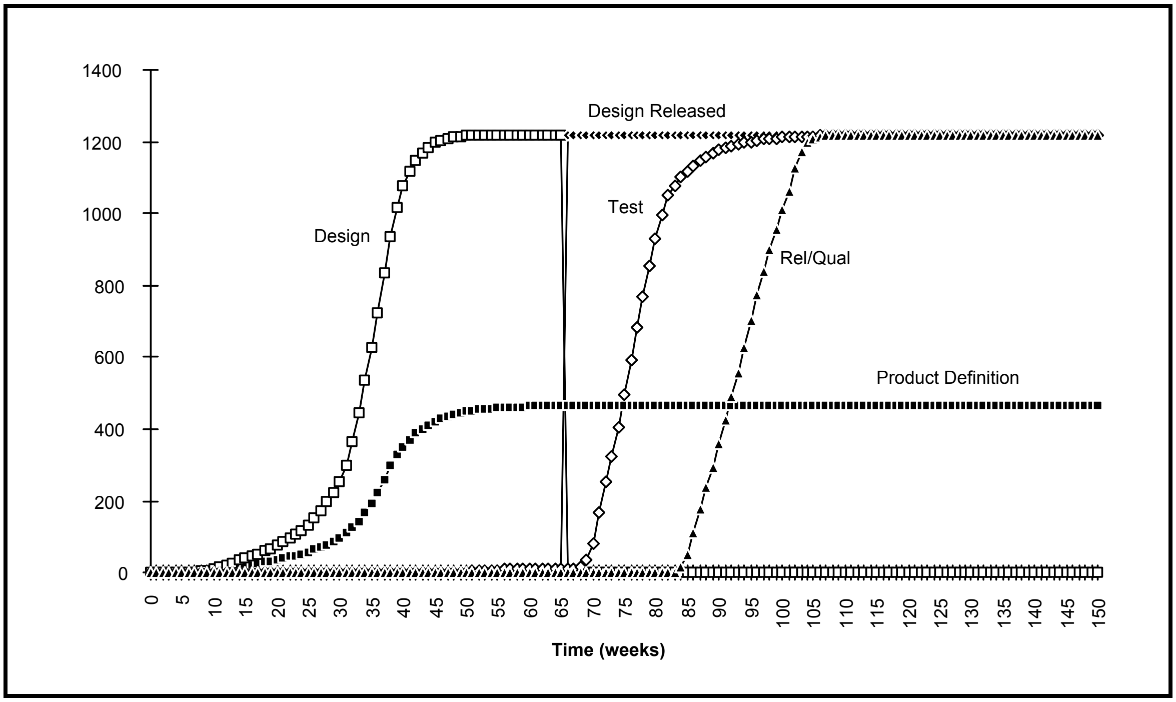

Figure 5-28: Project Simulation
No Coordination Policy Project

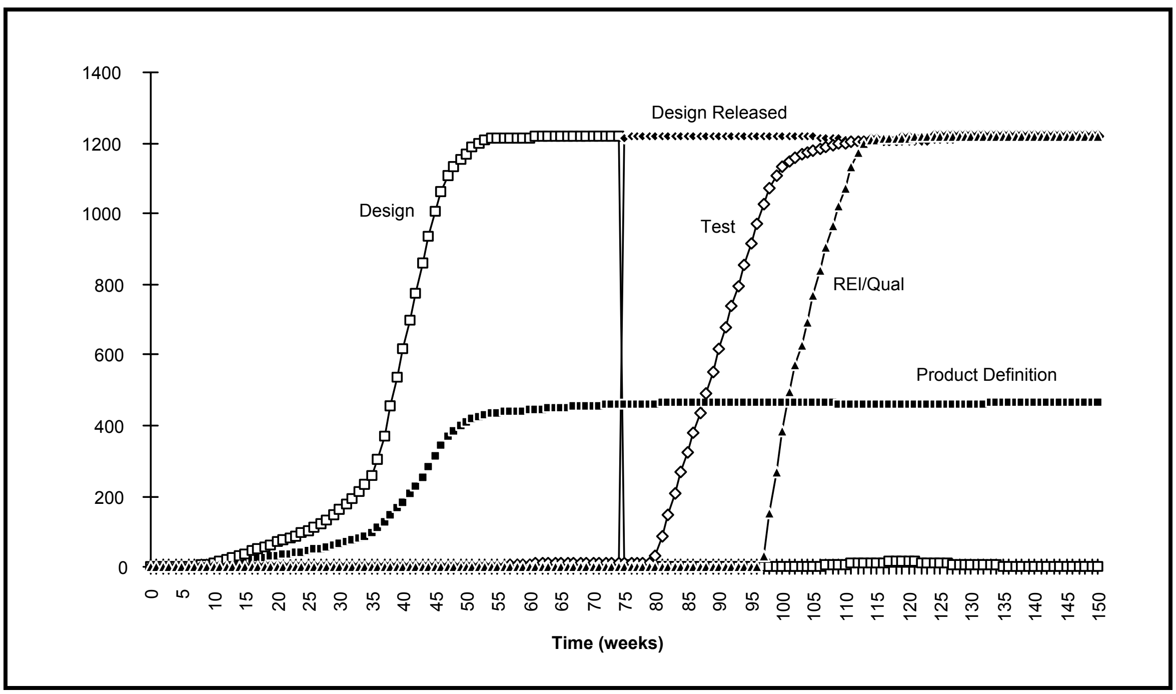

Figure 5-29: Project Simulation
Passive Coordination Policy Project

The slower completion of all phases and the later initial release of design tasks shown in Figure 5-29 indicate that the passive coordination policy finds some inherited errors.

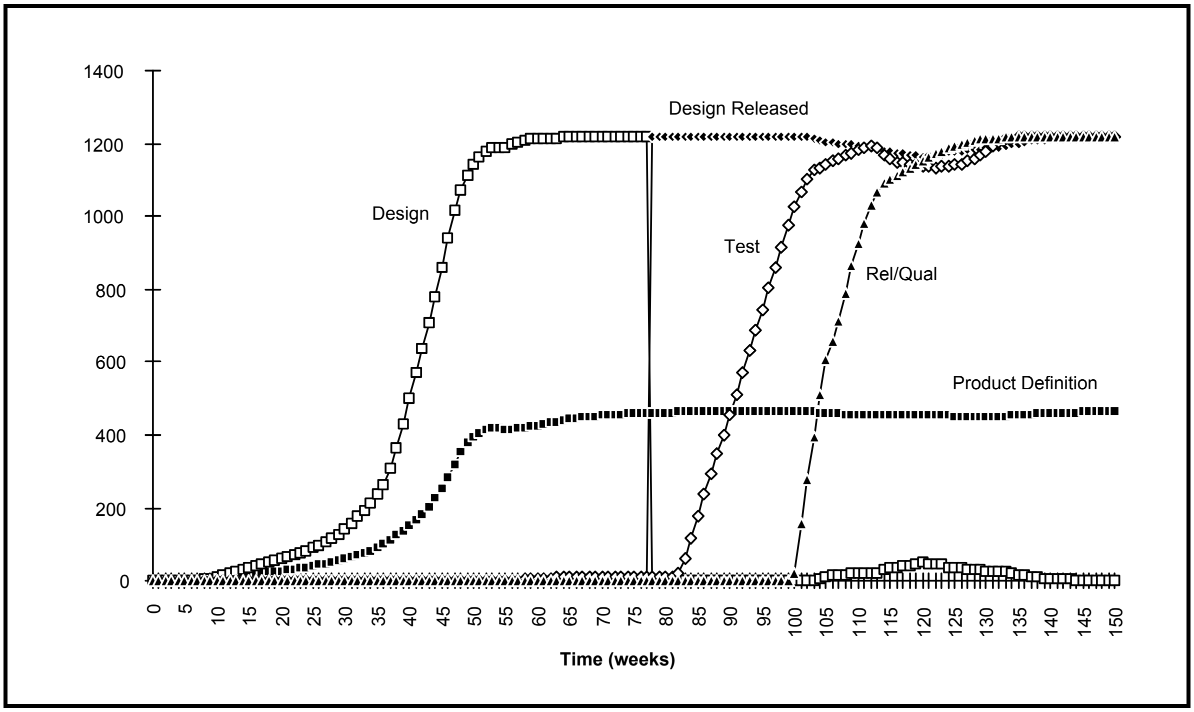

Figure 5-30: Project Simulation
Active Coordination Policy Project

The active coordination policy most clearly shows the impacts of inherited errors and coordination on Tasks Released phase behavior. The discovery of inherited errors from the Product Definition, Design and testing phases by the Reliability/Quality Control phase generate coordination and rework in all phases and delay project completion.

Table 5-5 shows the performance of the project under the three coordination policies.

Performance Measure	Coordination Policy *No Coordination*	 *Passive Coordination*	 *Active Coordination*
Cycle Time *(weeks)*	131	135	140
Quality *(percent defects released from final phase)*	54.4	53.2	46.9
Cost *(dollars X $1,000,000)*	1.016	1.076	1.203

Table 5-5: Project Performance Using Three Coordination Policies

5.6.4 Analysis of Project Behavior in Response to Coordination Policies

The project performance shown in table 5-5 does not confirm to the expectations of much of the literature concerning the impacts of coordination on project performance. Although the literature is not unanimous it generally expects all three performance measures to improve (i.e. measures in Table 5-5 decrease) as coordination increases. Table 5-5 shows a significant (14%) quality improvement but a degradation of cycle time and cost performance. Four points explain this discrepancy. First many of the conclusions of the existing coordination literature are drawn largely from a higher level of aggregation (e.g. Wheelwright and Clark, 1992 at the industry level). Research conclusions based on more aggregate data may be unable to identify differences in types of performance in individual projects. Research based on the project level of aggregation tend to be more limited in their expectations of coordination policies (e.g. Morelli and Eppinger, 1993). Second, some of the conclusions in the existing literature are based primarily on the opinions of industry practitioners (e.g. Cooper and Kleinschmidt, 1994 questionnaire-based research). Conclusions drawn primarily from the opinions of practitioners may reflect the current predictions or expectations of the business phases more than actual experience. Research based on objective data tend to be mixed (e.g. Iansiti, 1993). Third, these large amounts of aggregated and practitioner-based research suggest that coordination research is relatively young and holds an overly optimistic perspective of the potential of coordination to improve project performance[5.6]. Fourth, the project conditions assumed by the majority of the research may vary from those used in the analysis above.

Despite its differences with some of the literature the model performance shown in Table 5-5 makes intuitive sense. Coordination is an activity which does not directly contribute to the completion, evaluation, or correction of development tasks in the same direct way that basework, quality assurance, and rework do. Therefore a project with limited resources and more coordination could be expected to spend less resources on development tasks which release tasks and therefore take longer. If costs are related to cycle time, more coordination would also be expected to increase costs. The improvement in quality with increased coordination also makes sense if coordination is seen as a primary means of sharing knowledge within the development team and thereby influencing the generation and finding of errors.

But a deeper understanding of how coordination policies influence performance is needed to design robust coordination policies for improved project performance. Sensitivity analysis of the four coordination parameters used to describe coordination policies revealed that the Coordination Labor Delay and Coordination Priority had significant influence on performance while the Coordination Minimum Task Duration and Time to Adjust Expected Coordination Productivity had little impact. Project performance using both passive and active coordination policies were simulated across a range of labor delay and priority values to analyze their impacts. Cost performance was found to be reflected in cycle time performance. Therefore cycle time and quality will be the focus of this discussion. Project quality performance as Coordination Labor Delay and Coordination Priority vary is shown in Figures 5-31.

[5.6] I believe that coordination research is following a path similar to that of quality research in which an almost euphoric response to the discovery of a high leverage tool is followed by a slowly growing awareness of the pitfalls and complexity of the issue. Much of coordination research appears to still be in the early euphoric stage.

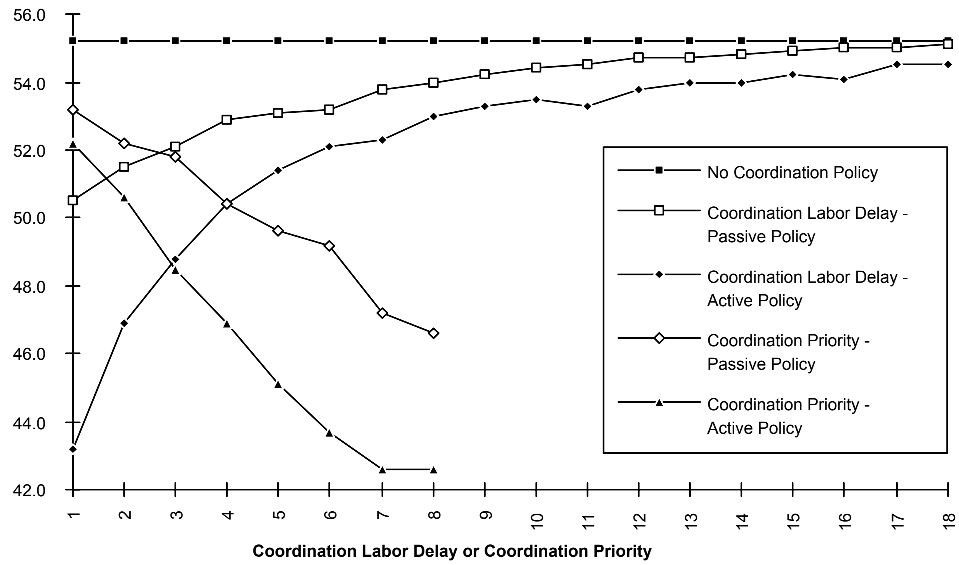

Figure 5-31: Project Quality Performance versus Coordination Labor Delay and Coordination Priority

Project quality decreases with increasing Coordination Labor Delay for both the passive and active coordination policies. The active policy produces better quality than the passive coordination policy. Project quality improves with increasing Coordination Priority for both the passive and active coordination policies. The active policy produces better quality performance and a slightly steeper improvement with increasing Coordination Priority. The project quality performance changes due to increasing labor delay and priority agree with initial intuition concerning the impact of project structure on system behavior. Increased labor delay is expected to create times when coordination labor is needed but not yet assigned to coordination tasks. This would allow more errors to be generated and released undiscovered. The result would be more released errors and lower quality as the coordination labor delay increases. Increasing coordination priority also generates expected behavior. More coordination would result in fewer errors being generated and more errors being discovered. Therefore the expected response to increasing Coordination Priority is improved quality.

Project cycle time performance as Coordination Labor Delay and Coordination Priority vary is shown in Figure 5-32.

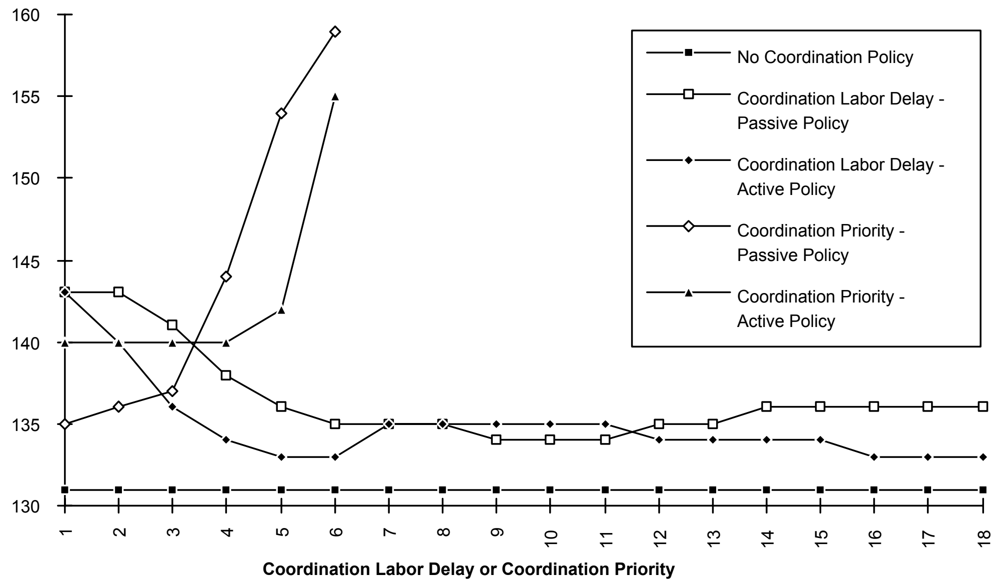

Figure 5-32: Project Cycle Time Performance versus Coordination Labor Delay and Coordination Priority

Project cycle time increases with increasing Coordination Priority in both the passive and active coordination policies. Both policies produce significant cycle time changes, although the project cycle time under the passive policy changes more than with the active policy case. Project cycle time behaves slightly differently as the Coordination Labor Delay increases under the passive and active coordination policies. Cycle time decreases as labor delays increase under both policies but the passive coordination policy allows the cycle time to increase again slightly as delays grow very long, whereas the active coordination policy continues a slow cycle time decline.

In contrast to quality performance, cycle time behavior is counterintuitive. Both increased Coordination Priority and increased Coordination Labor Delay are expected to increase cycle time. While increased coordination priority does this, increasing the delay in providing labor for needed coordination *decreases* project cycle time. Delays in shifting labor to a development activity in each development phase is intuitively expected to cause a delay in the completion of the entire project. But the model behaves very differently by producing decreasing project cycle time as the Coordination Labor Delay increases. Beyond the initial decrease in project cycle time the model behavior reverses itself with slightly increasing cycle times under the passive coordination policy. The next section uses the model structure to explain both the intuitive and counterintuitive behavior described above.

5.6.5 Causes of Project Behavior in Response to Coordination Policies

The model structure can be used to explain the model's behavior. Under the no coordination policy all available labor is used by basework, quality assurance and rework toward the release of tasks. This accelerates the project, reducing cycle time. But the lack of coordination increases the number of errors generated and decreases the number of errors discovered, reducing project quality. These causal links are shown in Figures 5-33, 5-34 and 5-35, which are described in more detail below. The project behavior under the no coordination policy is constant for all values of Coordination Labor Delay and Coordination Priority because the no coordination policy prevents the use of any coordination labor.

The explanation of the model behavior under the passive and active coordination policies begins with the feedback structure shown in Figure 5-33 which describes the intuitive behavior of the project's quality performance shown in Figure 5-31.

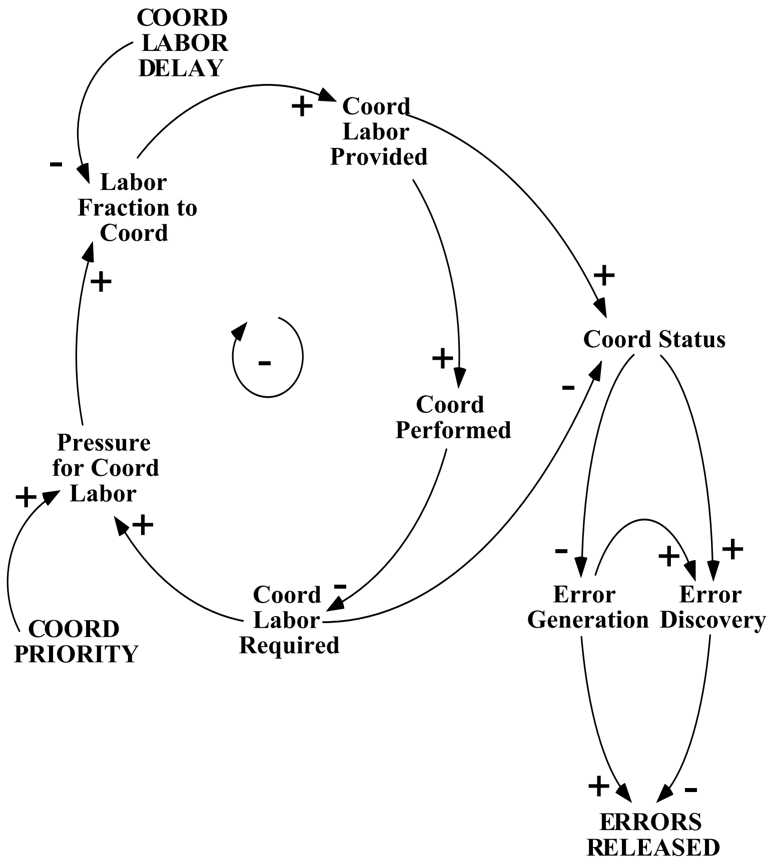

Figure 5-33: Causal Relationships Linking Coordination Labor Delay and Coordination Priority to Project Quality

The negative feedback loop in Figure 5-33 seeks to match Coordination Labor Provided to Coordination Labor Required by increasing and decreasing coordination labor in response to coordination demand. Increasing the Coordination Priority and decreasing the Coordination Labor Delay strengthen this loop, thereby speeding up the supply of coordination labor in response to a change in demand. The faster increase in labor increases the Coordination Status, which decreases the generation of errors and increases the percent of existing errors which are discovered. Both these influences decrease the number of errors released, thereby increasing project quality. Therefore increased Coordination Priority increases quality performance and increased Coordination Labor Delay decreases quality performance, as shown in Figure 5-31.

The causal loop diagram in Figure 5-34 explains the increase in cycle time with increasing Coordination Priority shown in Figure 5-32.

5-32).

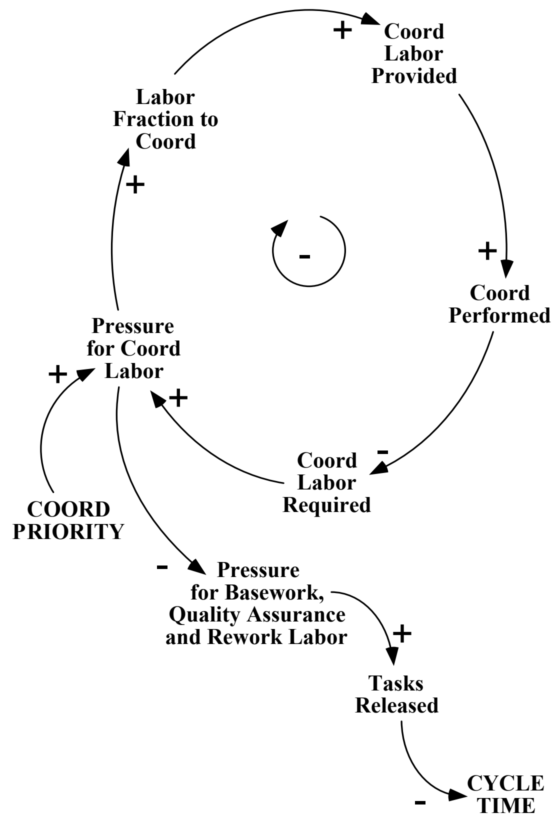

Figure 5-34: Causal Relationships Linking Coordination Priority to Project Cycle Time.

The Coordination Priority directly impacts the Pressure for Coordination Labor in the negative loop shown in Figure 5-34. This Pressure works with the analogous parameters for basework, quality assurance, and rework to determine which activities get the available labor. As the Coordination Priority increases more of the available labor is used for coordination and therefore less is available for basework, quality assurance, and rework. Since basework, quality assurance, and rework drive the release of tasks more directly than coordination "starving" those activities of needed labor increases cycle time. Increasing Coordination Priority effectively "steals" some of the limited available labor from basework, quality assurance, and rework for the coordination activity. Therefore increased Coordination Priority increases cycle time as shown in Figure 5-32.

Finally the Figures 5-33 and 5-34 can be combined and expanded to show the model structure which generates the model's counterintuitive behavior in which increasing Coordination Labor Delay decreases cycle time initially and eventually allows cycle time to increase slightly (Figure 5-32).

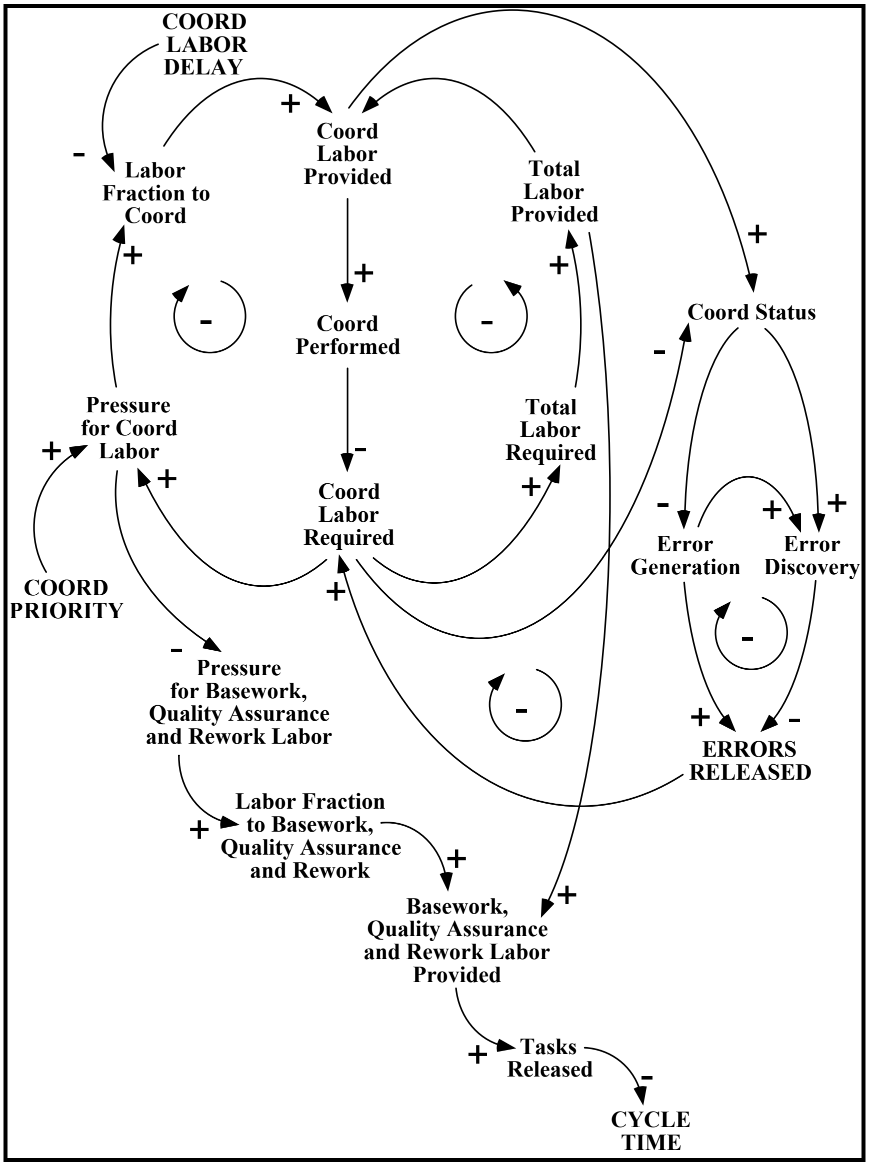

Figure 5-35: Causal Links from Coordination Labor Delay to Cycle Time.

The initial counterintuitive behavior (decreasing cycle time with increasing delays) can be explained and understood through the interaction of the two feedback loops shown in the upper left portion of Figure 5-35. The negative loop on the left side acts to shift available labor to the coordination activity in response to coordination need as described previously. When the left loop (the coordination labor allocation loop) is dominant labor is "stolen" from basework, quality assurance, and rework for coordination faster than the total labor pool can be increased. A strong coordination labor allocation loop allows the causal links in the lower left portion of Figure 5-35 to dominate, so when the coordination labor allocation feedback loop dominates system behavior cycle times are relatively high. The negative feedback loop on the right side increases the Total

Labor Required (for all four development activities) and thereby the Total Labor Provided (after significant delays). When the right loop (the total labor loop) is dominant the labor pool grows fast enough to meet the project's total labor needs without significant "stealing" of labor by coordination from basework, quality assurance, and rework. A strong total labor loop keeps Total Labor Provided high enough to not restrain basework, quality assurance, and rework from releasing tasks. When the total labor feedback loop dominates system behavior cycle times are relatively low. The total labor loop tends to eventually dominate behavior. This is because the "stealing" of labor only shifts the use of the available labor and does not alter the deficit between the Total Labor Required and the Total Labor Provided. Therefore the total labor loop seeks to fill the total labor need regardless of the allocation of labor among the four development activities and the labor allocation loop only temporarily controls the system behavior.

The system behavior becomes clearer when seen as a shifting of the dominance from the labor allocation loop to the total labor loop. Early in a project the labor allocation loop dominates when labor quantity is inadequate. This slows project progress by stealing labor from basework, quality assurance, and rework. But the total labor loop eventually provides the needed labor, allowing all four development activities to proceed unhindered by labor shortage. This tends to accelerate the project. The faster the dominance shifts from the labor allocation loop to the total labor loop, the faster the project will shift from expanding the project cycle time to reducing it. Short Coordination Labor Delays keep the labor allocation loop strong longer, delay the shift in loop dominance to the total labor loop, and therefore tend to produce projects with longer cycle times. As the Coordination Labor Delay increases the labor allocation loop becomes weaker, allowing the total labor loop to dominate earlier and the project cycle time to begin shrinking instead of growing sooner. Therefore increasing Coordination Labor Delays cause cycle times to decrease, as shown for Coordination Labor Delays from 2 to 9 weeks in Figure 5-32.

The slight increase in the cycle time as the Coordination Labor Delay grows under the passive coordination policy can be explained with the error generation and error discovery negative feedback loops[5.7] shown in the right portion of Figure 5-35. When the Coordination Labor Delay is long adequate quantities of labor provided by the total labor loop do not assure adequate coordination labor. This is because the long delay causes the labor allocation loop to be so weak that even when there is enough labor it is not allocated to the coordination activity. Under these

[5.7] For clarity each of these two loops represent several feedback loops linking error generation and error discovery to coordination labor. All are negative loops with similar impacts on system behavior.

conditions the lack of coordination labor allows the Coordination Status to fall, increasing the number of errors generated and decreasing the errors discovered. The result is an increase in coordination (and therefore the total) work required and a resulting increase in cycle time. Since coordination labor is typically only a small portion of the total labor for the project the increase in the cycle time is minimal. This represents a second shift in loop dominance from the total labor loop to the error generation and error discovery loops. Therefore very long Coordination Labor Delays tend to increase cycle times as shown for Coordination Labor Delays above 11 in Figure 5-35. This second shift in loop dominance does not occur under the active coordination policy because the level of coordination does not drop low enough to allow the error loops to dominate.

The preceding hypothesis that the timing of a shift in loop dominance controls the change from increasing to decreasing cycle time can be tested with the Product Development Project Model. The hypothesis is supported if a faster shift in loop dominance from the coordination labor allocation loop to the total labor loop reduces project cycle time. The timing of the shift can be influenced by altering the strength of the total labor loop. If cycle times decrease as the total labor loop strengthens then the hypothesis is supported. If cycle times do not decrease as the total labor loop strengthens then a different explanation for the counterintuitive behavior shown in Figure 5-32 should be sought.

The strength of the total labor loop will be varied by changing the maximum Headcount Adjust time constant parameter in the development phases. Figure 5-36 shows cycle time as the total labor loop is strengthened.

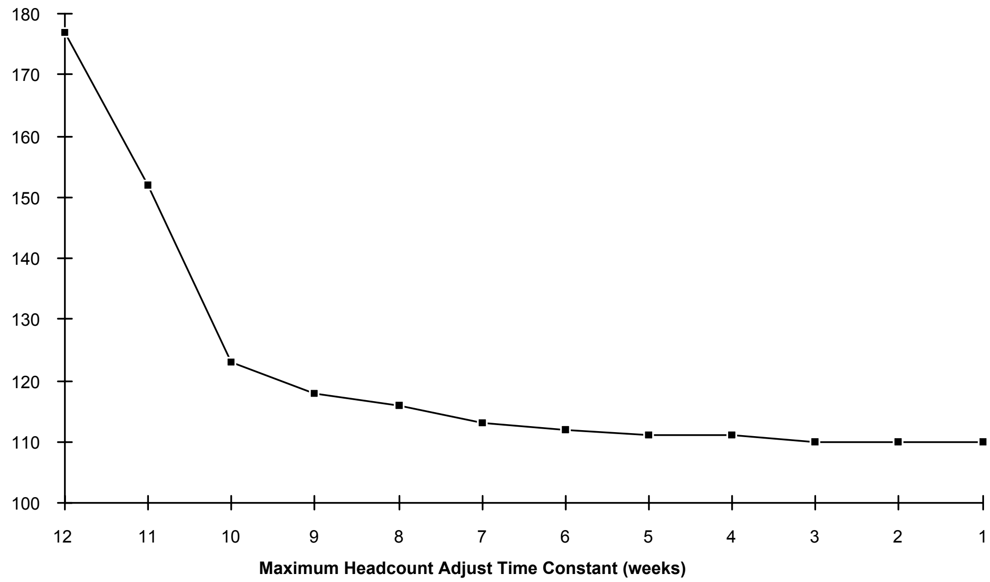

Figure 5-36: Coordination Policy Hypothesis Test

The test using the Product Development Project Model shows that cycle time drops as the Headcount Adjust Time decreases (total labor loop strengthens). This supports the hypothesis by showing that faster shifts to the total labor loop decrease cycle time.

5.6.6 Impacts of Coordination Policy Analysis on Coordination Policy Design

The differences between the performance due to using no coordination, a passive coordination policy or an active coordination policy shown in Figures 5-31 and 5-32 indicate that the design of a project coordination policy influences project performance. The analysis indicates that the relationships among coordination policy parameters and performance are neither linear or monotonically increasing or decreasing. These relationships require a more sophisticated understanding of project dynamics than simple rules-of-thumb such as "decrease delays to accelerate a project" to design effective and robust coordination policies. The shifting of dominance among feedback loops provides a means of understanding the impact of structure on behavior.

One implication of the analysis is that more coordination may improve some performance measures while degrading others. This places the policy designer in the common position of being forced to trade improved performance in one domain (e.g. cycle time) for poorer performance in another (e.g. quality) (Rosenau and Moran, 1993; Rosenthal, 1992). The relative values of performance in the time, quality, and cost domains are specific to development products, industries, and organizations. For example the Python project valued cycle time performance very highly and development cost performance less. In contrast many building development projects value development cost more than the other two measures. These considerations are essential to determining realistic performance targets prior to designing a coordination policy. Typically some compromise among the three performance measures is a realistic and desirable project target.

However other impacts of coordination policies may generate improvement in all three performance measures. For example increased coordination may reduce the time required to add new labor through early warning of work loads. Productivities may increase significantly due to increased knowledge and development team morale. Figure 5-37 shows a simulation in which these additional beneficial effects of a coordination policy are included. In comparison to the No Coordination policy results in Table 5-5 this coordination policy improves all three performance measures: cycle time (from 131 weeks to 125 weeks), quality (from 54.4% errors released to 43.5% errors released) and cost (from $1,016,000 to $893,000).

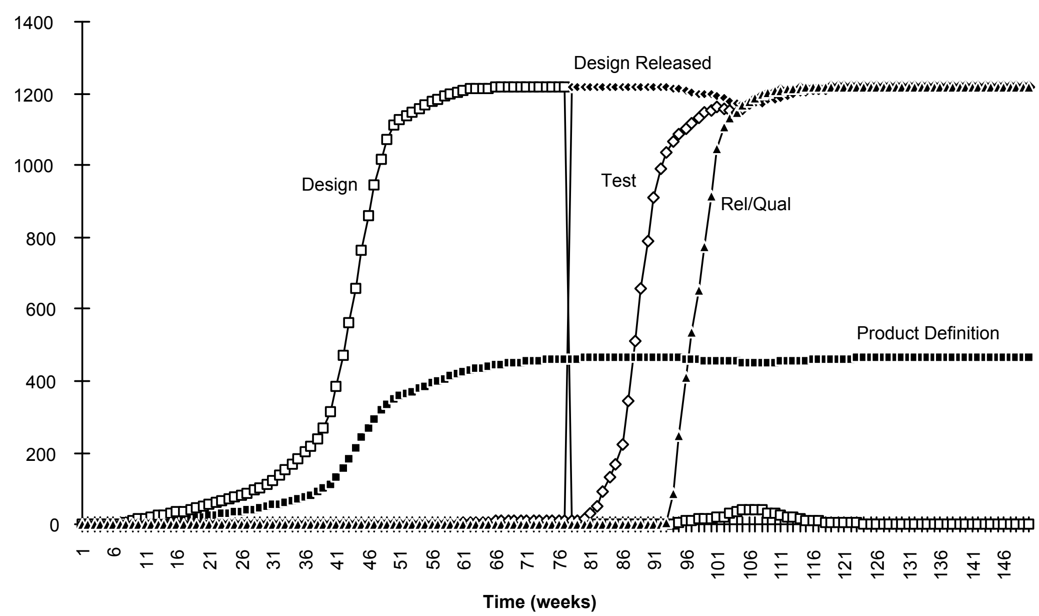

Figure 5-37: Project Simulation of A Coordination Policy with Additional Effects Included

A traditional approach to designing a coordination policy based on the structures described might seek to control the strength of the coordination labor allocation loop to reduce the cycle time delays caused by starving basework, quality assurance and rework of labor. This might entail selecting a compromise between low Coordination Priority and long Coordination Labor Delay to minimize the strongest negative impacts of both caused by their extreme values. However a broader perspective can improve coordination policy design and potentially project performance beyond that possible with a traditional approach. Such an expanded approach could use the understanding of shifting loop dominance to identify and manipulate other high leverage points in the coordination system. For example an alternative to weakening the coordination labor allocation loop to accelerating the shift in loop dominance from the coordination labor allocation loop to the total labor loop could be sought. One alternative is to strengthen the total labor loop. This could be done by shortening the delays in the total labor loop. An example of such a delay is the Headcount Adjustment Time parameter which slows the change in total labor in response to total labor needs. A policy of predicting project labor needs before they actually occur and beginning the search for appropriate personnel could significantly influence this parameter. The model structure can be beneficially used to identify and investigate a variety of coordination policies which utilize project parameters and relationships beyond those which strictly define coordination activities in the most narrow sense. These investigations can also reveal parameters which may limit the effectiveness of a policy. An example of such a limiting parameter is the Maximum Headcount, which could constrain the effect of reducing the Headcount Adjustment Time parameter. Another advantage of basing coordination policies more broadly is the increased robustness of such policies. An example can be seen in the ability of the active coordination policy in the analysis which activated more coordination parameters to restrain growth in cycle time as Coordination Priority increased when compared to the passive coordination policy which engaged fewer coordination parameters.

5.6.7 Policy Analysis Summary

The analysis of coordination policies using the Product Development Project Model revealed that the impacts of increased coordination differ among the three project performance measures and from much of the current coordination literature. Three coordination policies were simulated: no coordination, passive coordination, and active coordination. As coordination policies became more active project cycle time and cost performance decreased and quality improved. The Coordination Labor Delay and the Coordination Priority were found to be the most influential of the four parameters used to describe coordination policies. Changes in project performance as Coordination Labor Delay and Coordination Priority increase were simulated and revealed counter intuitive cycle time behavior. The model structure was used to explain both the intuitive and counter intuitive behavior. The results of the analysis indicate that coordination policy designers must consciously distinguish among relative values of performance in different domains to set realistic performance targets. The analysis also points to the use of the model structure and shifting loop dominance to expand the search for effective coordination policy parameters and relationships for effective and robust policies.

5.7 The Python Development Project Summary

The Product Development Project Model was calibrated to the Python project, a semiconductor chip development project consistent with the model assumptions. Data concerning the theoretical and practiced development process and organization were integrated into the model structure and parameter estimations for calibration. This required the addition of a model structure describing the aggregation and holding of completed and checked tasks before their release as a group. Two model configurations were calibrated to the Python project. A single phase configuration was calibrated to the design phase of the Python project and a multiple phase configuration was calibrated to the majority of the Python product development process. The calibrated model simulations reflected the fundamental behavior modes of the reference modes of the Python project. The shifting of loop dominance within the model structure provided explanations for the model behavior and expanded the understanding of how the high leverage points of the system identified in chapter 3 influence system behavior.

The model was used to analyze coordination policy impacts on performance measures. Four parameters were used to describe coordination policies. Three coordination policies were tested. As coordination became more active quality improved but cycle time and cost performance degraded. This is intuitive but not in agreement with much of the coordination research literature. The Coordination Labor Delay and Coordination Priority were shown to have significant influence on performance. As these parameters increase quality responded as expected but cycle time decreased with increasing Coordination Labor Delays. The model structure was used to explain both the intuitive and counterintuitive behavior of the system. The coordination policy analysis can have significant impacts on the design of coordination policies and project management. The design of effective and robust coordination policies requires relative valuation of different project performance measures and the setting of realistic targets. Shifting feedback loop dominance can provide an expanded understanding of the relationships between structure and performance and thereby lead to improved coordination policy design.