Chapter 6

Conclusions

6.1 Recapitulation

Successful product development projects are critical to competitiveness in several industries. Changing competitive forces such as globalization, increased customer sophistication, and accelerating technology are increasing the difficulty and leverage of managing product development projects. In response many firms have replaced their functional sequential development paradigm with a more cross-functional concurrent approach. The new paradigm increases the impacts of the relationships among project components on performance and thereby the influence of dynamic project features such as feedback, delays and nonlinear relationships. Successful management of these projects requires an understanding and use of the dynamics of projects. Existing research has focused on a static view of project management, especially concerning the impacts of process structure, where the Critical Path Method paradigm has dominated for decades. This research investigates the impacts of dynamic project structure on performance with a focus on the influences of the development process.

A dynamic simulation model of a multiple phase project was built using the system dynamics methodology. The model integrates several previously developed and tested project structures and adds a separate structure for the development process. Simulations describe the behavior generated by the interaction of customized development phases and a project management structure. Each phase explicitly models the impacts of development processes, resource capacity, scope, and targets on four development activities: basework, quality assurance, rework, and coordination. Project performance is measured in time, quality and cost. The model structure is based on previous project models and field data from a practicing product development organization. Sensitivity tests indicate that the three performance measures are most sensitive to error generation rates and process description parameters. A signal processing model of a small portion of the project model was built and used to investigate the similarities and differences between the system dynamics methodology and a more traditional modeling approach.

The model was calibrated to a computer chip development project for a single development phase configuration and a four phase configuration which represented the majority of the development process. Quantitative and qualitative data concerning the development organization, process, and project was collected for parameter estimation. Project and phase behavior and performance data were collected and analyzed to generate reference modes. Testing revealed that when the model is appropriately parameterized the resulting simulated behavior closely resembles the actual historical behavior of the project. The similarity in behavior modes between the project behavior and model simulations support the model's ability to simulate development project dynamics.

The model was applied to the investigation of coordination policies for improved project performance. Three coordination policies representing different levels of coordination were represented in the model. Model simulations indicate that increased coordination improves quality but can degrade cycle time and cost performance. The model structure helps explain the causes of this behavior. Analysis of the influences of two descriptors of coordination policy reveal that cycle time can decrease as the delay between coordination labor need and coordination labor provided increases. The model structure helped identify the timing of a shift in feedback loop dominance from the allocation of limited labor among development activities to the provision of adequate labor for all development activities as the cause of this counterintuitive behavior. The use of loop dominance as an alternative to single-link open-loop rules of thumb for describing and managing projects was discussed.

6.2 Major Findings and Discussion

6.2.1 The Role of Development Process in Project Behavior

The characteristics of development processes significantly impact the dynamic behavior of projects. Processes can directly influence project progress by constraining development work. Development processes can also influence progress indirectly by setting the total and relative demands for the labor used in different development activities. The influence of the development process can dominate the influences of some resource structures. In addition, a development process structure which includes recycling of work in a closed loop flow, available work constraints within and between phases and minimum task durations was able to describe complex process impacts on project behavior. The identification of these structures and their dynamic impacts indicate that project models should include dynamic descriptions of development processes.

Based on this finding I speculate that previous dynamic project models which reflect actual behavior have incorporated development processes into resource structures and parameter estimates. Another (albeit unlikely) possibility is that the development process of the projects modeled had no significant impact on behavior. It is possible to describe process constraints which are so loose and resource, scope, and target constraints which are so strict that the process has little influence. However the significant impact of the process structures in the Product Development Project Model and the differences between the process influences and those of other project components indicate that development processes should be modeled explicitly and separately from resources, targets, or scope to distinguish process impacts.

6.2.2 The Role of Feedback, Delays, and Nonlinear Relationships in Project Behavior

The research shows that projects have many potentially important feedback loops. Some are closed loop flows of work or errors in which pieces of the project leave a position or condition in the project through the performance of development work and return to the condition for repeat performance of the development work. Many other potentially important feedback loops return information about project conditions for use in decision making. These feedback flows of work, errors, and information are dynamic and critical to describing the causal relationships within a project which drive behavior.

Development processes and other project features do not move instantaneously or without bias. Understanding the size and character of the delays which alter these flows is important in relating project structure to behavior. Changing those delays can be a potentially effective tool for improving project performance.

Several of the important relationships which drive project behavior are nonlinear. In particular the relationships which describe the available work constraints within and between phases can be described in significantly greater depth with nonlinear relationships than with linear approximations. These improved descriptions expand the range of project relationships which can be modeled.

6.2.3 Project Constraints

Projects have many constraints on their behavior which resist extremes in performance. The sensitivity analysis produced no fluctuations in project performance over 200%. This is small compared to the performance of many complex systems such as corporate growth or profits which can experience much larger variations. This characteristic of the model is at least partially due to the large number of negative feedback loops which can influence project behavior. The more negative feedback loops in a model or system the more likely that one of those loops will be activated or become dominant and redirect project behavior when behavior approaches extremes. Another partial explanation is that the size of the system is limited by the number of tasks in the project. As modeled by the Product Development Project Model a project's scope cannot exceed the sum of the phase task lists. This constraint could be relaxed if (for example) the discovery of errors generated more rework than the correction of the flawed task.

6.2.4 An Important Gap in Project Management

There is a gap between the primary methods currently used to describe, model, communicate and manage projects and the complexity of the structures which drive the behavior of those projects. Most project models do not include the impacts of feedback, delays, and nonlinear relationships in the evolution of projects. Those that include some of these project features do not describe the internal structure of a project in adequate detail to identify significant causal relationships. Current project management practice is based on open-loop, single-link linear causal relationships which can be and often are reduced to lists of rules-of-thumb guidelines. The parameters and causal relationships identified by the Python developers is a specific example. Thomsett (1990) is an example of this perspective in the literature for practitioners. These tools are incapable of capturing the dynamic project behavior described by the Product Development Project Model and illustrated by the Python project. Additional tools are needed to extend the bounded rationality of project researchers and managers beyond the capabilities of currently available tools to include dynamic project features, structures and behavior.

Based on this finding I speculate that the gap between the tools currently used to understand and manage projects and the complexity of projects will keep a majority of the existing knowledge concerning managing project complexity trapped in the minds of experienced project managers and will restrain the development of new knowledge. The lack of tools for describing that knowledge prevents its testing, improvement, and communication to others. The gap has become a major cause of project performance problems as competitive forces and the new development paradigm increase project complexity. This finding also implies that, barring reduced project complexity, project performance cannot significantly improve without the development of tools for the description, modeling, and management of complex systems. One of the challenges of bridging this gap is the need for the tool to be useful for description, modeling, communication and management, not just one or two of these functions. The next major finding points toward a possible solution.

6.2.5 System Dynamics as a Tool for Researching Projects

The system dynamics methodology and its adjacent tools such as causal loop diagramming can describe project complexity. The Product Development Project Model is an example of such a description. The similarity of its simulation of the behavior of the Python project to field data support its ability to adequately describe dynamic impacts of complex causal relationships. When combined with the previous finding this indicates that the system dynamics methodology is a potential tool for bridging the gap between current project tools and project complexity.

While this research has shown that system dynamics can fill at least part of the research portion of the gap between current tools and project complexity it has not shown that it can currently bridge the entire gap. The methodology has proven itself successful as a tool for investigation and learning but has been applied more narrowly to communication about and management of projects. The use of changes in loop dominance in the policy analysis portion of this research to explain counterintuitive project behavior identifies it as a potential tool for bridging from system dynamics models to effective and efficient communication about project complexity. I speculate that using system dynamics to describe project complexity will increase the demand for explanatory and management practice tools such as changing loop dominance. I further speculate that improved understanding of loop dominance will facilitate meeting that demand.

6.3 Implications for Project Management

The research supports several previous insights about projects such as potential tradeoffs among performance measures and has revealed new insights about the dynamics of projects. Those new insights include:

- Development process dynamics can play as important a role in project behavior as project resources, targets and scope. The dynamic impacts of many project features exceed the ability of traditional project models to describe the impacts of structure on behavior.

- Different development activities within a single development phase influence project behavior differently. The characteristics which distinguished basework, quality assurance, rework and coordination made qualitative differences in project behavior. The distinction between activities which directly impact the completion and release of work and development activities which indirectly influence those activities is particularly important.

- The sizes of delays in dominant project processes have large impacts on project behavior. For example the largest minimum task duration for basework, quality assurance, or rework can constrain progress as development phases increase work rates. Delays in quality assurance processes have particularly large influences due to the many feedback loops which they impact.

- Development processes can trap work within and among development phases with many side effects which cause additional work and degrade performance. High error generation and discovery rates cause many iterations, thereby increasing total work and generally degrading project performance.

- Some project subsystems such as project targets include compensating feedback loops which weaken the effect of attempts to control project behavior. Several of these subsystems describe human reactions to project conditions and management policies.

These insights indicate that development practitioners need to understand and use feedback, delays and nonlinear relationships in the management of development projects. This requires an expansion of the project models used by practitioners. Without ignoring the project features used in traditional project models an expanded project model will focus on different types of project features as shown in Table 6-1.

Project Feature	**Traditional Project Model**	**Expanded Project Model**
Project Structures described	Static	Static & Dynamic
Causal Relationships	Open-loop	Feedback
	Single-link	Connected chains of links

Basis of Development Phase Descriptions	Time (phase duration)	Work (quantity of development tasks)
Basis of Inter-Phase Relationships	Temporal differences among phase starts and completions	Available work constraints Impacts of Changes
Schedule Management focus	Degree of Activity concurrence	Delay sizes and locations
Quality Management focus	Error prevention Iteration reduction	Flows of iterative work
Resource Management focus	Total resources available to project	Allocation of resources within project

Table 6-1: Management Focus in Traditional and Expanded Project Models

The research also implies that improvements in project performance will be limited if dynamic issues are not addressed. This research supports the existing literature in identifying increased project management difficulty with increased concurrence of development activities. Pressure for improved project performance (particularly reduced development cycle time) are expected to continue the current move toward increased concurrence. The resulting management challenges are primarily dynamic.

6.4 Contributions

6.4.1 A New Research Tool

The Product Development Project Model provides a first attempt to integrate into a testable framework the development process and decision making features of development projects. The result is a new and valuable research tool for investigating the dynamics of projects. The model represents an initial description of how primary project structures interact to dynamically impact project performance. By relating process, resources, scope, and target features to performance the model plays a similar role for dynamic project features as the Critical Path and PERT methods provide for static project features. The model does this by integrating many existing project components into a single project model, introducing and testing several new dynamic project structures, and building a flexible project model. The new structures include:

- **Explicit and separate descriptions of the development process, including:**
  a) The available work precedence relationships describing constraints due to processes both internal and external to each development phase. These relationships allow much deeper constraint descriptions than available in previous models.
  b) The description of the development process with four explicit and separately modeled activities allows improved development phase descriptions.
  c) The minimum task durations allow the unique description of each of the four development activities in each phase.

- **Development process as a generator of demand for development activities:** This is the first primarily demand-driven system dynamics project model. The demand for each of the four development activities directly drives the process limits to progress through the minimum task durations and indirectly drives the resource limits to progress by setting goals for resource structures. In previous models progress is constrained by a resource bottleneck which enlarges and shrinks due to many factors but does not seek any goal. In this model resources seek to fill a demand for each activity set by the development process. This requires the modeling of the concept that all incomplete work may not be available for development work at any given time and that project developers and managers respond only to the current demand for development and not all future development needs. This differs from previous system dynamics models of projects which assume that all incomplete tasks are available.

- **Closed loop flows of flawed tasks:** These structures are the next step in an evolution of project work model structures which grew out of solely resource based models. This model contributes an explicit stock of work waiting to be corrected with in the closed loop flow of work and the development process descriptions to the work flows.

- **Coflow structures for flawed work:** These structures allow the separate modeling of errors instead of using errors to alter a single flow of work. This allows more explicit and detailed modeling of the causes of error generation and discovery and their impacts.

- **Two-directional flow of error information between phases:** Error information is passed in both the downstream and upstream directions. This impacts both the receiving phases through work corruption and coordination demand and the error generating phase through the return of errors for rework and coordination. This error information passing is used within a flexible dependency network.

- **Generic project structure:** A flexible project model structure allows the modeling of many different types of projects. This is accomplished primarily with two model features. First, a flexible number (up to five) of generic linked development project phases can be customized with parameter values to reflect different development phases. Second, a structure for describing the dependency network among those phases directs the flows of work and errors. Previous system dynamic models of projects have had fixed numbers of phases of fixed relationships among them.

6.4.2 Project Dynamics Insights and a Tool for Product Development Practitioners

The insights described in the "Implications for Project Management" section above are also contributions of this research. They illustrate the need for tools which facilitate the expansion of project models by practitioners to include dynamic issues. The Product Development Project Model is one such tool. The Model can help practitioners improve development project practice by improving the understanding of project dynamics in several ways:

- Small portions of the model can be used to investigate the generic impacts of project structures and changes in project parameters. For example ICI has spent significant time and money to accelerate the checking of development work for errors (reduce their quality assurance minimum task duration). ICI's product development process improvement engineer has used a small version of the model to improve his understanding of the impacts of additional quality assurance task duration reductions in a in a context of multiple process and resource constraints.

- The model can be calibrated and used to improve understandings of the impacts of specific project subsystems on project performance. The investigation of the coordination subsystem in chapter 5 is an example of this type of application.

- The model could be calibrated to specific product development operations and used to design and analyze project management policies. This work could be the basis for the development of improved project management heuristics which include dynamic impacts.

- The model can be revised to focus on a specific type of dynamic behavior and developed into a "management flight simulator" suitable for facilitating learning about the dynamics of projects by a group of product developers and managers.

6.5 Limitations of the Research

The model is designed and built to represent a class of problems (development projects). The variety of projects within that class will always require model calibration to realistically reflect specific projects. The limitations of this model specified in this thesis suggest important issues for the broader application of the model and its underlying concepts within the class of development projects.

- **Model size:** The size and resulting complexity of the model will tend to increase as the model is applied to larger projects. This can be partially addressed by increasing the model's level of aggregation. However this may obscure project features of interest.

- **Level of aggregation:** The level of aggregation of the model and its focus are related. The level of aggregation will tend to increase with project size. Too high a level of aggregation may hide the causal relationships which generate the behavior of interest. In contrast a higher level of aggregation will cause some small variations (e.g. the temporary headcount drop in the Python project) to become irrelevant. Too low a level of aggregation unnecessarily increases the modeling effort and potentially introduces misleading model features. The purpose of the model application and the resulting focus will indicate an appropriate level of aggregation.

- **Data collection for parameterization:** Both larger and smaller projects can raise important data issues. Larger projects will increase the number of phases and sources of data. How to effectively collect and integrate that data from potentially different forms into usable information must be addressed. This challenge may be partially ameliorated by more formal data collection and documentation procedures used in larger projects. Smaller projects tend to generate challenges in collecting data for which no formal or documented trail exists. This requires additional judgment concerning the role of project components and expanded methods of data collection.

- **Organizational and development culture boundaries:** Development projects which span organizational and cultural boundaries can generate issues concerning how the different organizations and cultures interact which are not addressed here. These issues can be very important in development projects (for example see Ward, 1995) and should be accounted for in the application of the model to projects with significantly different or separate organizations and cultures.

- **Environmental change:** Changes in the project environment can also be significant in development projects. Technology development which precedes product development is an example (Iansiti; 1992, 1993a, b, c, d). Development organization support (Roberts; 1964) and competition among projects (Weil et al.; 1973) for resources may also require additional model structure or special attention to model data.

6.6 Future Research

The findings and limitations of this work point to potentially valuable extensions. They include the investigation of :

- Relative sizes and types of influences of process, resources, targets, and scope subsystems on performance for specific groups of development projects based on industry (e.g. construction, automobiles, etc.), project size or number of phases.
- Relative sizes and types of influences of different development activities on project performance.
- Dynamic impacts of project features and policies on important non-performance measures such as project manageability or developer moral.
- Impacts of more detailed modeling of developer experience levels, types of labor and other aggregated project features
- Relax the model boundary assumptions to include multiple projects, market introduction and product performance, technological and organizational evolution, or market competitors.
- Add model structure to internalize currently exogenous inputs to the model such as resource availability, process descriptors and development activity priority.

6.7 Summary Conclusions

This research addressed the important issue of the causes of dynamic behavior in product development projects by building, testing and applying a dynamic simulation model of a multiple phase project. Feedback, delays and nonlinear relationships were found useful in describing the drivers of dynamic behavior. The concept of product development as a set of interactive demand-driven activities was used to build rich descriptions of causal relationships based on previous research and field data. The strong direct and indirect influences of development processes were identified by explicitly separating development processes from resources, scope, and targets. The use of model structure to explain project behavior was illustrated by applying the model to a specific type of project management policy. This identified changes in loop dominance as a potentially valuable tool in communicating the impacts of complex structure on behavior.

The research identifies a gap between current project models used for management and the complexity of project structures. A failure to bridge this gap is expected to limit project performance improvement. Expanding the knowledge and understanding of project dynamics is a critical part of meeting this need. The development of new or improved tools for communication and management practice is also expected to be essential to translating improved knowledge and understanding into improved project performance.

This research has contributed insights concerning the dynamics of projects, a tested framework for modeling projects based on demand for development activities, a tool for future research and a tool for improving the understanding of product development practitioners. This work has created opportunities for expanding the study of project dynamics in several potentially valuable directions.

This research has pushed project management toward a broader image of projects and its role in project performance. It points to ways of improving performance through improved understanding of project structure and behavior. Future research will expand and refine the understanding and use of dynamics to manage projects. But the foundation for extending project models to include fundamental dynamics exists today.