System dynamics applied to project management: a survey, assessment, and directions for future research

James M. Lyneis[a]* and David N. Ford[b]

James M. Lyneis is a Professor of Practice at Worcester Polytechnic Institute and Senior Lecturer at MIT, where he teaches both system dynamics and project management. Prior to his return to teaching, he worked for 25 years in the Business Dynamics practice of PA Consulting Group (formerly Pugh–Roberts Associates). His research interests include applications of system dynamics to project management, business strategy, and economics.

David N. Ford is a Professor in the Construction Engineering and Management Program in the Zachry Department of Civil Engineering at Texas A&M University. He teaches and researches project dynamics and the strategic planning and management of development projects. As an engineer in practice, Dr Ford designed and managed government, commercial, and residential development projects. He received Bachelor's and Master's degrees in Civil Engineering from Tulane University and a PhD from MIT in Dynamic Engineering Systems.

Abstract

One of the most successful areas for the application of system dynamics has been project management. Measured in terms of new system dynamics theory, new and improved model structures, number of applications, number of practitioners, value of consulting revenues, and value to clients, "project dynamics" stands as an example of success in the field. This paper reviews the history of project management applications in the context of the underlying structures that create adverse dynamics and their application to specific areas of project management, synthesizes the policy messages, and provides directions for future research and writing. Copyright © 2007 John Wiley & Sons, Ltd.

Syst. Dyn. Rev. **23**, 157–189, (2007)

Context

Projects abound in industry, public service, and many other endeavors. As a series of activities or tasks that (1) have a specific objective (scope) to be completed within certain specifications (requirements); (2) have defined start and end dates; (3) have funding limits; and (4) consume and/or utilize resources (Project Management Institute, 2000), projects have proven challenging to plan and manage. This is largely because project conditions and performance evolve over time as a result of feedback responses, many involving nonlinear relationships, and to accumulations of project progress and resources. This has made the application of system dynamics to project management a fertile and productive field of study. This paper surveys the large body of system dynamics work on projects, evaluates its progress, and suggests directions for future development.

Many different types of models have been developed to improve project management. These models include some of the system features and characteristics addressed by system dynamics. For example, basic project models such as the critical path method explicitly model causally linked development activities and phases and cost control models use forecasted performance gaps (e.g., budget deficits) to allocate funds. More advanced models, such as the computational models developed by Levitt *et al.* (1999) and others, are quite system dynamics-like, as they include linked development activities as well as feedback. Another body of work models multiple projects, using system

[a] PO Box 121, Weston, VT 05161, U.S.A. Email: jmlyneis@alum.mit.edu
[b] Zachry Department of Civil Engineering, Texas A&M University, College Station, TX, 77843-3136, U.S.A.
* Correspondence to: James M. Lyneis.

Received February 2007; Accepted June 2007

System Dynamics Review Vol. 23, No. 2/3, (Summer/Fall 2007): 157–189
Published online in Wiley InterScience
(www.interscience.wiley.com) DOI: 10.1002/sdr.377

dynamics as well as other approaches. Surveying all of these works is beyond the scope of a single article. Therefore, we focus here on models of single projects built using the system dynamics methodology. But even models of single projects are too numerous to describe their structures or applications in detail. Therefore, in this article we focus on the most important and general model structures in conceptual form, and provide references to additional details. Our work is based primarily on the published literature and our experience using system dynamics to model projects. In particular, we describe contributions resulting from work we have done that has not, and very likely never will be, published or otherwise made available.

The literature on system dynamics models of projects varies widely in the level of detail provided, especially in model structure descriptions, from complete model disclosure to almost none. Some authors focus on model structure while others focus on model use and describe model structure only in general terms (e.g., the "Strathclyde" work of Ackerman *et al.*, 1997; Eden *et al.*, 1998, 2000). Our assessments are necessarily limited when model equations or detailed structure information is not available. However, our review reveals a direct and positive relationship between the access provided by authors to model details and the subsequent use of those models by other researchers and practitioners.

The remainder of the paper is structured as follows. Important conceptual model structures are described in a way that relates them to system dynamics principles and in the approximate chronological order of development. Model structures are followed by some typical project behaviors they produce. The paper then discusses applications, policy lessons, and future research directions organized by traditional areas of project management, and finishes with a general assessment of the work to date and suggestions for future development.

Structures underlying project dynamics

The structures that system dynamicists have used to model projects can be described in four groups based on the central concept that they integrate into project models. The categorization provides a meta-structure of project model structures and relates those structures to the system dynamics methodology. The four model structure groups are:

1. *Project features:* System dynamics focuses on modeling features found in actual systems. In projects these include development processes, resources, managerial mental models, and decision making. Modeling important components of actual projects increases the ability to simulate realistic project dynamics and relate directly to the experiences of practicing managers.
2. *A rework cycle:* System dynamics has a set of canonical structures that drive much of the dynamics of specific model types. The inventory-WIP structure

Published online in Wiley InterScience
(www.interscience.wiley.com) DOI: 10.1002/sdr

in supply lines (Sterman, 2000) and the aging structure in *Urban Dynamics* (Forrester, 1969) are examples. The canonical structure of system dynamics project models is the rework cycle.

3. *Project control:* Modeling, analyzing, and improving the control of dynamic systems is the objective of applying system dynamics in many domains. Since project managers seek to deliver on time, on budget, and with the quality and specifications required, modeling the controlling feedback loops through which management attempts to close gaps between project performance and targets directly applies one foundation of system dynamics to project management.
4. *Ripple and knock-on effects:* Policy resistance and unintended consequences are fundamental explanations used by system dynamics for many adverse behaviors. "Ripple effects" is the name commonly used in projects to describe the primary side effects of well-intentioned project control efforts. Modeling ripple effects in projects captures and leverages the concept of policy resistance. "Knock-on effects" refers to the secondary impacts of project control efforts, i.e., the impacts of ripple effects, often caused by processes that produce excessive or detrimental concurrence or human factors that amplify the negative effects via channels such as morale. Capturing knock-on effects in project models uses the concept of unintended side effects to explain project behavior and performance.

Project features

Projects almost always consist of a collection of tasks that are performed in parallel and in series. Therefore, a principal feature of all system dynamics project models is the representation of development tasks or work packages as they flow through a project. Development tasks typically start in a stock of tasks *to be done* and then flow through the project's development process until the stock of tasks *done* reaches the level of project completion. In a model of a specific project, tasks in the development process may represent the entire project, or may be disaggregated into more detailed development phases (e.g., into developing specifications or coding software). Another feature of projects represented in system dynamics models is the application of resources to manage the flows in the development process, based on management's perceptions of project conditions.

Roberts (1964, 1974) developed the first published model of a project and introduced the flows of project work in terms of "job units" based on resources applied and productivity. In addition, he introduced several important concepts that represent management's understanding of project conditions: (1) perception gaps—differences between perceived progress and real progress, and between perceived productivity and real productivity; and (2) underestimating scope and effort required. These errors can cause under- or misallocation of resources that ultimately feed back to affect project performance.

Published online in Wiley InterScience
(www.interscience.wiley.com) DOI: 10.1002/sdr

Roberts was followed by a succession of modelers who improved the richness of project models by adding other features found in actual projects, including both development processes and management. Improved representations of development processes included, but are not limited to: distinguishing work done correctly from work done incorrectly (first by Pugh–Roberts Associates (PRA),[1] Cooper, 1980, and Richardson and Pugh, 1981); multiple project phases (first by PRA, Cooper, 1980); separate effort for quality assurance (first by Abdel-Hamid, 1984); nonlinear constraints of work availability on progress (first by Homer *et al.*, 1993); development projects as value-adding aging chains (first by Ford and Sterman, 1998a); and concurrence constraints limiting how much work can be done in parallel (Ford and Sterman, 1998a; Madachy, 2002).

Simultaneously, modelers were improving project models by adding features that reflect the human aspects of projects, especially project management features and processes such as the "freezing" and "unfreezing" of designs due to changes and uncertainties (Strathclyde in Ackerman *et al.*, 1997; Eden *et al.*, 1998, 2000), releasing completed work to downstream phases (Ford, 1995), using contingency funds (Ford, 2002) and schedule buffers (Park and Pena-Mora, 2004), and resource allocation policies (Joglekar and Ford, 2005). These features clearly exploit the power of system dynamics to model human decision making, such as modeling decisions driven by gaps, delays in human processes, and nonlinear relationships. Most formulations of these features apply traditional structures described in other system dynamics literature.

The rework cycle

The rework cycle is, in our opinion, the most important single feature of system dynamics project models. The rework cycle's recursive nature in which rework generates more rework that generates more rework, etc., creates problematic behaviors that often stretch out over most of a project's duration and are the source of many project management challenges. PRA developed the first rework cycle model, shown conceptually in Figure 1 (Cooper 1980, 1993).[2] In this form the rework cycle includes four stocks of work. At the start of a project or project stage, all work resides in the stock "Original Work to Do". Progress is made by applying effort. A fraction of the work being done at any point in time contains errors. Work done correctly enters the "Work Done" stock and never needs rework (unless later changes render that work obsolete). However, work containing errors enters the "Undiscovered Rework" stock. Errors are not immediately recognized, but are detected as a result of doing downstream work or testing. This "Rework Discovery" may occur months or even years after the rework was created. Once discovered, the backlog of "Rework to Do" demands the application of additional effort. Reworking an item can generate or reveal more rework that must be done. Therefore, some reworked items flow through the rework cycle one or more subsequent times.

Published online in Wiley InterScience
(www.interscience.wiley.com) DOI: 10.1002/sdr

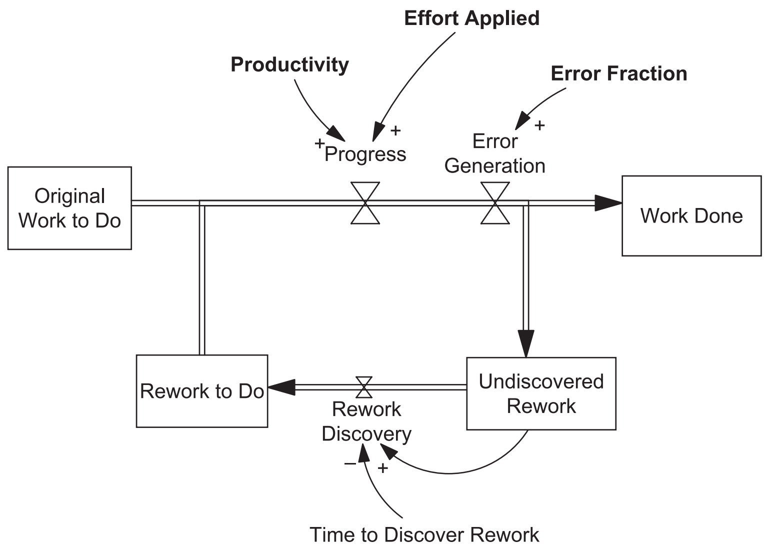


Fig. 1. The rework cycle (adapted from Cooper, 1993)

Subsequent modelers have developed other rework cycles, principally Abdel-Hamid (1984) and Ford and Sterman (1998a, 2003b). They retain the rework cycle's recursive nature, but add other features or use other model structures. For example, Ford and Sterman's aging chain structure moves work through a series of backlogs and improvement activities that initially complete, then test, and then release work with the rework cycle linked to the aging chain at the Quality Assurance backlog. This structure uses a separate quality assurance effort and adds parallel rework cycles in co-flow structures to distinguish between errors that are generated within a phase and those generated by upstream phases. Other authors, such as Park and Pena-Mora (2003), elaborate on the work flows and distinguish between rework to correct flawed work (e.g., removing and replacing poor construction) and rework initiated to respond to externally generated changes. The importance of the rework cycle is indicated by the fact that all known system dynamics project models subsequent to PRA's original work have included a rework cycle.

Controlling feedbacks

In modeling controlling feedback, system dynamicists have focused on the information processing of project managers. Project performance is typically measured in terms of schedule, cost, quality, and scope. Management actions to control a project's performance are modeled as efforts to close the target–performance gap in one or more performance dimensions. The two basic methods available to practicing project managers have been modeled: move

Published online in Wiley InterScience
(www.interscience.wiley.com) DOI: 10.1002/sdr

project behavior closer to targets (e.g., work overtime), or move targets toward project behavior (e.g., slip a deadline). Both methods use negative (controlling) feedback loops, with managerial responses typically being proportional to gap size. However, limits often exist on the size and speed of adjustments and both methods impose costs (monetary and other types). Project targets are often set for future dates (e.g., cost when the project is completed), and therefore modelers have often included managerial forecasting of performance. As noted above, system dynamicists have consistently modeled perceived conditions separately from actual conditions, with the former driving project control actions and the latter driving actual progress. In several models the structures used to model perceived conditions reflect managerial mental models and are not just delayed actual conditions. For example, managers generally include undiscovered rework in work believed to be completed, and therefore overestimate progress. This, combined with reporting systems that often estimate productivity based on work believed to be done to date divided by hours spent to date, can overestimate progress early in the project and underestimate it later (e.g., Ford, 1995; Lyneis *et al.*, 2001). As will be discussed, these generate adverse feedback effects in the form of ripple effects.

Controlling to meet a deadline is common in project management practice and has been a particular focus of many system dynamics models of projects. We therefore use it here to illustrate a model of managerial action for project control. Three common actions can be taken to correct a situation in which project managers forecast that they will miss a deadline: (1) hire additional workforce (most project models starting with Roberts, 1994); (2) work overtime (PRA models, Ford and Sterman, Strathclyde); and (3) work faster (PRA models, Abdel-Hamid, Strathclyde). As indicated in Figure 2, these form the "Add People", "Work More", and "Work Faster/Slack Off " feedback loops. In these loops, an expected completion delay, as indicated by more time required to finish the work remaining than the time remaining to the project deadline, initiates hiring, overtime (which increases effort via more hours per worker), higher intensity of work (which increases productivity; e.g., output per person-hour), or a combination. In isolation these actions increase progress, reduce the remaining work, and thereby reduce the expected completion delay. Note, however, that if the expected completion delay is zero or negative (i.e., the work remaining seems doable in less than the time remaining), work intensity is often reduced, thereby creating the "Slacking Off" variation on the "Work Faster" loop—work intensity and productivity decrease and work remaining does not fall as fast as originally planned. Another variation on this "Slacking Off" loop, not shown for clarity, is a "gold-plating" loop whereby slack in the project leads designers to add "unnecessary" features and capabilities, thereby eliminating the slack. Another possible action, slipping the deadline, is indicated by the negative loop in the lower right of Figure 2. Deadline slip is often taken only as a last resort when the adding resource loops fail to completely solve the problem.

Published online in Wiley InterScience
(www.interscience.wiley.com) DOI: 10.1002/sdr

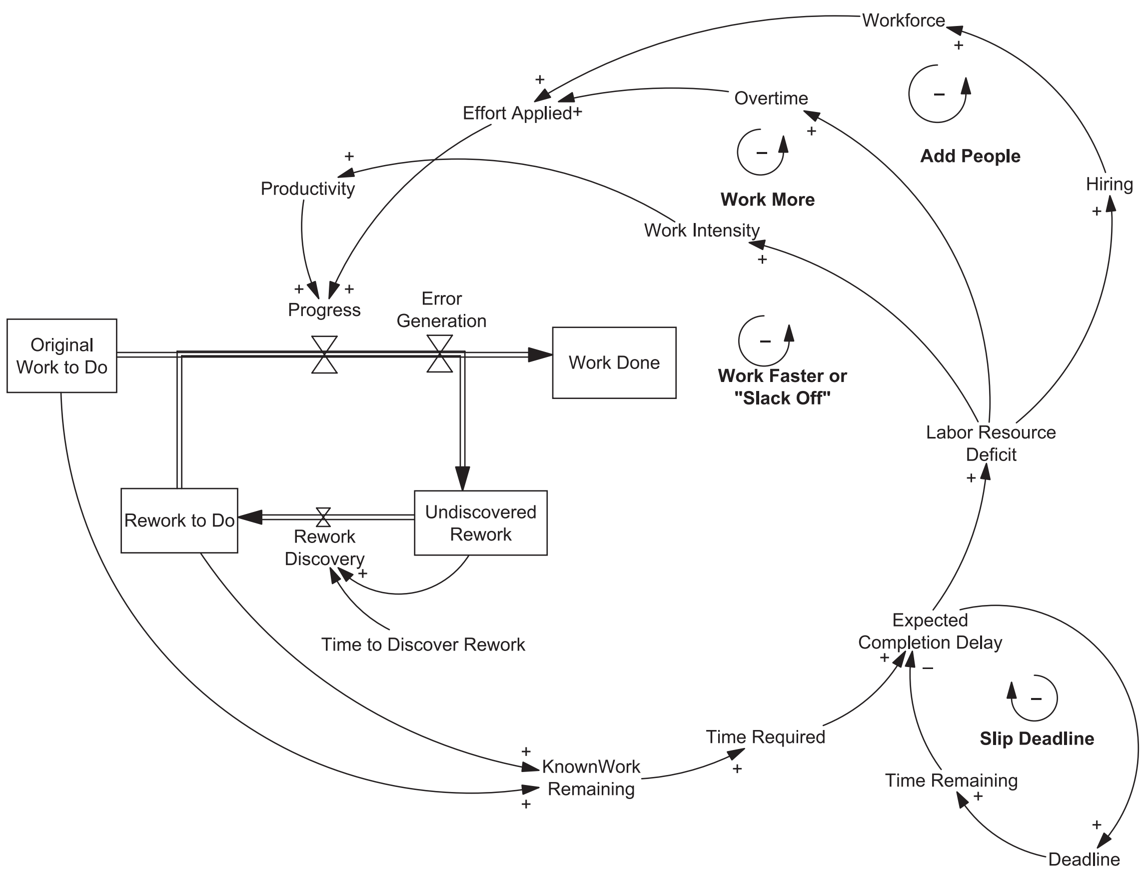


Fig. 2. Controlling feedback loops for achieving a target schedule (deadline)

Ripple effects

Unfortunately, actions taken to close a gap between project performance and targets have unintended side effects that generate policy resistance. These ripple effects are the primary impacts of project control on rework and productivity. Figure 3 adds four important ripple effect feedbacks of the three project control actions shown in Figure 2. These effects typically reduce productivity or quality (by increasing the error fraction and rework). Hiring can dilute experience as workers with less skill and/or less familiarity with the project are brought on, and because they require experienced developers to divert time to training instead of doing development (most models since Roberts). Larger workforces can increase congestion and communication difficulties, which increase errors and decrease productivity (PRA, Abdel-Hamid, and

Published online in Wiley InterScience
(www.interscience.wiley.com) DOI: 10.1002/sdr

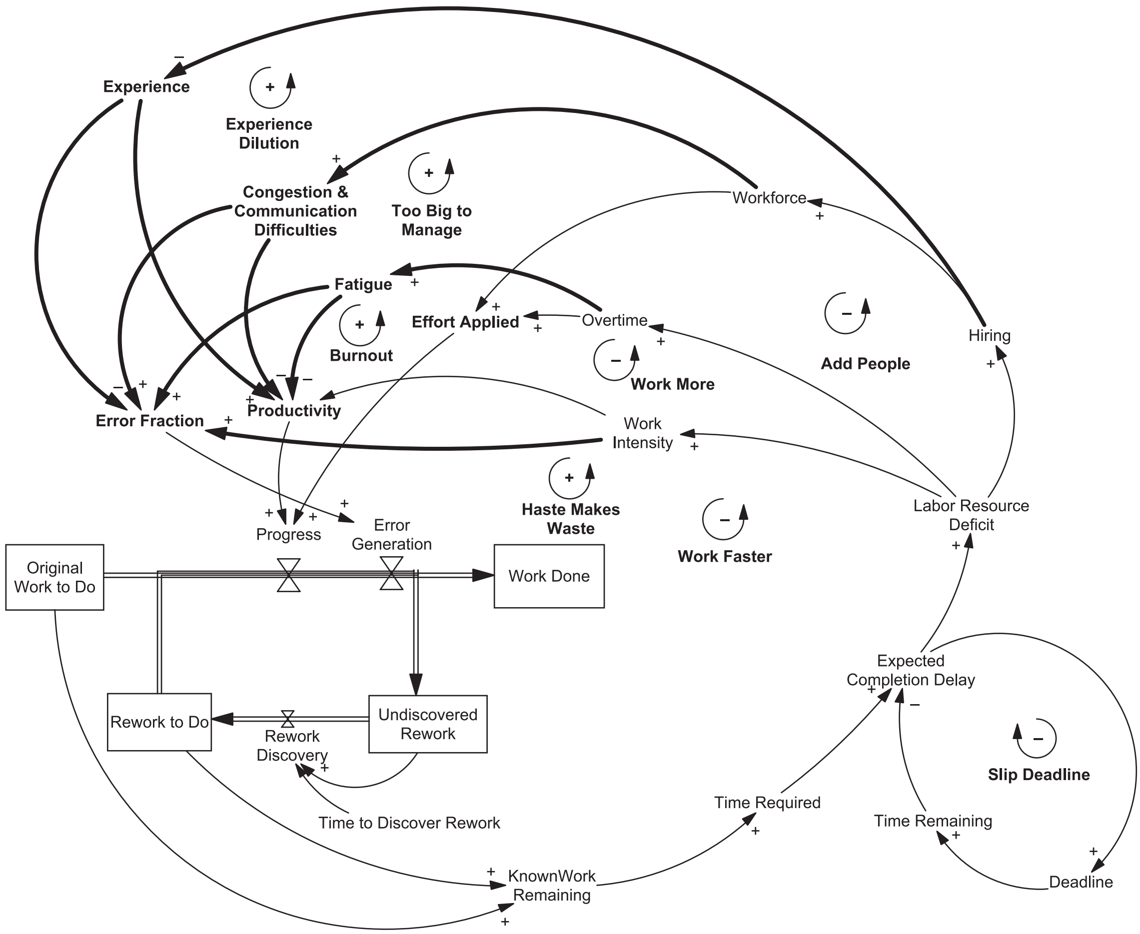


Fig. 3. Policy resistance via ripple effects of rework and controlling feedback to improve schedule performance

Ford–Sterman models). Overtime leads to fatigue (after a delay) that also increases errors and decreases productivity (all models that include overtime). Higher work intensity increases errors (PRA models, Abdel-Hamid, Strathclyde). Reduced productivity and increased rework keeps the amount of work remaining greater than it would have otherwise been, thereby increasing labor resources needed to finish on time. These effects form the Experience Dilution, Too Big to Manage, Burnout, and Haste Makes Waste loops. Consistent with system dynamics theory, they are reinforcing loops which can cause a project to spin out of control. While these ripple effect feedbacks are characteristic of many of the early project models, they were usually not clearly diagrammed or highlighted by authors. Some of these loops appear to be explicitly diagrammed for the first time in Sengupta and Abdel-Hamid (1993) and Cooper (1994).

Published online in Wiley InterScience
(www.interscience.wiley.com) DOI: 10.1002/sdr

Knock-on effects

Ripple effects generate secondary and tertiary feedbacks; some are consequences of physical processes related to work flow through projects that propagate from upstream work to downstream work, both within a phase of work (e.g., design), and between phases of work (e.g., from design to construction), while others are due to "human" reactions to project conditions. Many of these effects are generated by the activation of ripple effects structures described above. Figure 4 builds from Figure 3 to illustrate that these "knock-on" relationships can generate significant harmful dynamics, including:

- "Haste creates out-of-sequence work"—trying to accomplish more tasks in parallel than physical or information constraints allow, whether by adding

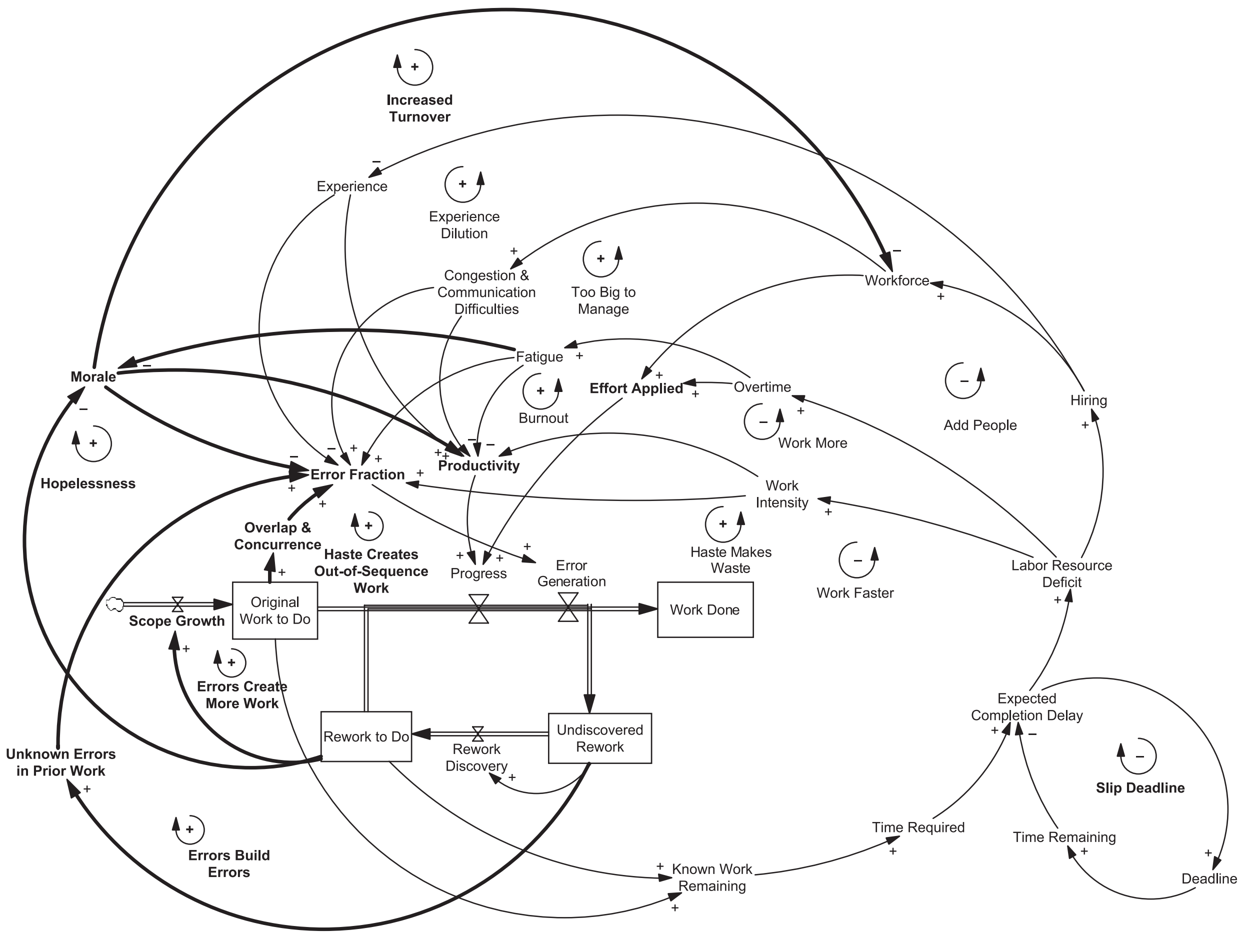


Fig. 4. Policy resistance via "knock-on" effects to controlling feedback to improve schedule performance

resources or exerting schedule pressure, can cause work to be done concurrently, out of the desired sequence, or both. This reduces productivity and increases errors (PRA models, Ford and Sterman, 2003b; Cooper, 1994; Lyneis *et al.*, 2001).
- "Errors build errors"—undiscovered errors in upstream work products (e.g., design packages) that are inherited by downstream project phases (e.g., construction) reduce the quality of downstream work as these undiscovered problems are built into downstream work products. Coded software is a good example of this contamination effect (PRA, Abdel-Hamid, Ford-Sterman, and Strathclyde models, Ford *et al.*, 2004; Lyneis *et al.*, 2001).
- "Errors create more work"—the process of correcting errors can increase the number of tasks that need to be done in order to fix the problem, or can increase the work required because fixing the errors takes more effort than doing the original work. Taylor and Ford (2006) demonstrate that this feedback can create "tipping point" dynamics through which fraction complete can stop increasing and begin to decline, often resulting in project cancellation.
- "Hopelessness"—morale problems can exacerbate the effects—fatigue and rework can create a sense of "hopelessness" that increases errors and reduces productivity, and which also increases turnover (PRA and Strathclyde models).

Finally, while the primary adverse ripple and knock-on feedbacks as typically modeled by system dynamicists are internal to the project (often including suppliers and subcontractors), adverse feedbacks through clients and customers can initiate or amplify internal project dynamics (Rodrigues and Williams, 1998; Reichelt, 1990; McKenna, 2005). Examples of these external actions include the following:

- Clients often change scope or requirements, activating project control actions, ripple effects, and knock-on effects, thereby degrading projects that were otherwise successful.
- Projects which are under-budgeted can lead to efforts by the contractor to increase the budget via change orders, which divert efforts from other project work.
- Poor schedule performance and slipping of deadlines can reduce client trust in the project team, with the resultant demands for more progress reports; more time spent on progress reporting and interacting with the client reduces productivity, slows progress and necessitates additional schedule slip through a reinforcing loop.
- Reduced client trust can also lead to reluctance by the client to tolerate further deadline slippage, which increases schedule pressure and aggravates project control problems.
- In the extreme, if project problems lead to litigation (while the project is still ongoing), then diversion of management attention to litigation activities

Published online in Wiley InterScience
(www.interscience.wiley.com) DOI: 10.1002/sdr

can reduce attention to the project itself, and thereby exacerbate project performance.

These "external" feedbacks are sometimes included in project models.

Assessment of system dynamics project model structures and research needs

Based on our project management experience and modeling, we believe that the development of project models has now captured the majority of the important features of development projects: the characteristics of the rework cycle, controlling feedback loops, and ripple and knock-on effect re-enforcing loops. While specific models differ in their level of detail with regard to the phases of work represented, the complexity of the rework cycle, and the feedback effects that are represented, formulations of these processes have been developed and are documented for others to use.

There are, however, two areas of structural research we feel warrant additional work:

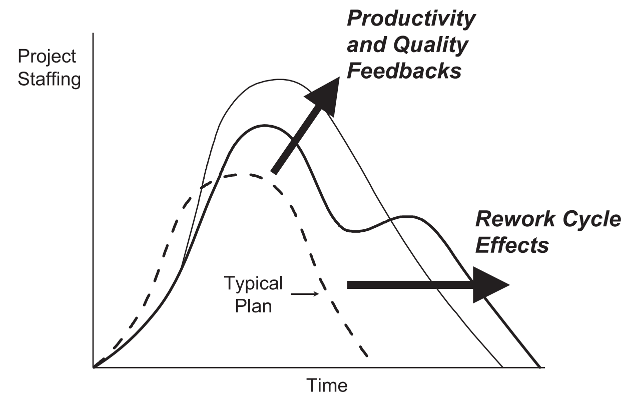
1. Nearly all the ripple and knock-on effect feedbacks manifest themselves through nonlinear relationships. There is relatively little discussion of the nature and strength of these relationships and, in particular, how they might differ by phase of work (e.g., design vs. construction) or by type of project (e.g., software vs. hardware), or as a result of changes in process and tools (e.g., CAD systems might reduce the strength of errors on error feedback, and make error fraction less sensitive to people factors), and how different strengths may alter any policy heuristics. Ford and Sterman (1998b) provide one approach and examples, but much more work is needed.
2. While nearly all system dynamics project models represent aspects of the ripple and knock-on effects of project controls to achieve project performance targets, the secondary consequences of adjusting targets has not been investigated as deeply. While some modelers have represented slipping schedule as well as adding resources, and sometimes compute a value for the damages of late delivery, they rarely explicitly examine the secondary impacts of such slips on performance of the product in the market.

Common project behaviors

The most common behavior of actual projects cited in the literature is failure to meet performance targets (for examples, see Lyneis *et al.*, 2001). System dynamicists have used the project structures described above to explain these failures and suggest improvements. Figure 5 illustrates typical (but by no means all) possible behaviors for project staffing: *planned* staffing often builds up to a peak, and then gradually declines; *actual* staffing, however, can deviate

Published online in Wiley InterScience
(www.interscience.wiley.com) DOI: 10.1002/sdr

Fig. 5. Some rework cycle and productivity/quality effects on project staffing dynamics



significantly from the plan. Often the ramp-up of staff is delayed, then overshoots the planned peak and remains high longer (sometimes with a second hump). Such projects typically experience both schedule and budget overruns, and also deliver projects with reduced scope and lower quality than desired or required.

How do the above feedback structures contribute to this behavior? The ramp-up of staff is often delayed either because of external conditions such as late finish of other projects, but also because management overestimates productivity and underestimates true project scope and the delays in getting resources. The "hiding" of undiscovered rework further delays the recognition of staffing needs. As the need for additional staff is recognized and controlling actions taken, ripple effects and knock-on effects on productivity and quality of managerial responses tend to increase effort required and therefore staff needs, causing project staffing and labor hours to peak higher and later than planned. Cycling of work through the rework cycle also pushes project completion later in time.

The evolution of the fraction of work completed is also often used to describe progress on real projects. As illustrated in Figure 6, a period of relatively steady apparent progress is often followed by a period of slow progress before completion. This behavior results initially because of underestimates of true work scope and the hiding of undiscovered rework, and later as managerial responses initiate ripple and knock-on feedbacks which slow progress by reducing productivity and increasing errors. In the end, progress is constrained by the discovery and correction of the last bits of rework through the rework cycle (often called the "90% syndrome"). Some researchers have documented projects that also experienced a slower initial start-up period that formed an elongated "S" behavior mode for progress (Reichelt and Lyneis, 1999; Ford, 1995; Ford and Sterman, 2003b). This basic behavior mode is sometimes augmented by periods of little or no net progress (i.e., project is "stalled", often

Published online in Wiley InterScience
(www.interscience.wiley.com) DOI: 10.1002/sdr

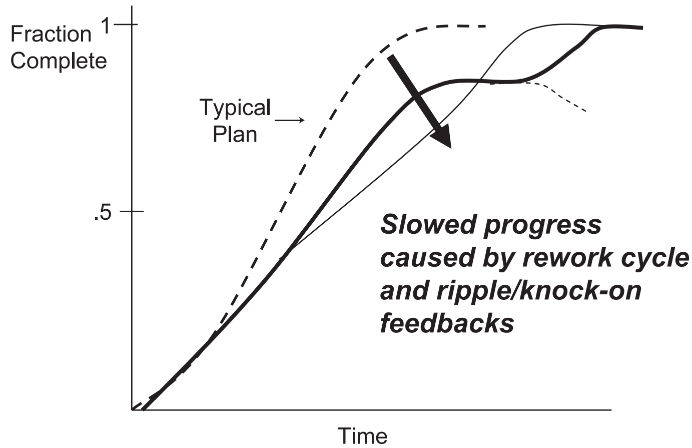


Fig. 6. Some rework cycle and productivity/quality effects on fraction complete

because of the recognition of undiscovered rework and actions to execute that work in a timely fashion), or by temporary (Ford, 1995) or permanent (Taylor and Ford, 2006) declines in net progress (i.e., a "decaying" project) because of added work to execute rework.

In the next section, we discuss what system dynamicists have learned regarding how managers can improve project performance, and what further work should be done to improve the practice of project management.

Applications and policy insights

Summary/overview

The work of several of those cited above has spawned significant follow-on work by numerous other researchers, consultants, and companies. The number of real-world applications is particularly significant. By our count, more than 50 companies have used system dynamics for project management on at least one project, and some companies on many projects. PRA alone are known to have applied system dynamics to over 100 projects. Together with the efforts of other organizations, therefore, the total number of such applications most likely exceeds 200 and continues to grow.

While these applications have been in a wide range of industries (aerospace, automotive, civil construction, energy and software to name a few), none are known to require major deviations from the basic structures described above. The usefulness of these structures across so many applications makes them as fundamental to system dynamics as "classic" structures such as the supply

Published online in Wiley InterScience
(www.interscience.wiley.com) DOI: 10.1002/sdr

chain described by Forrester (1961) and the infection, commodity, and growth structures described in Sterman (2000). However, while the basic structures of projects across industries have many dynamic similarities, projects in different sectors and industries have some unique features. Hardware projects differ in some respects from software projects; consumer goods differ from defense projects; one-of-a-kind, first-of-a-kind projects differ from product development projects. In most cases, researchers and consultants have adapted one of the generic models to a specific application area; in other cases, notably software and civil construction, a stream of research and application has occurred such that more specific versions of the structures underlying project dynamics, and specific policy issues, have been developed for these industries. While some of this work has general applicability and will be referenced below, a more complete listing and summary of the work is contained in an extended bibliography of system dynamics applied to project management accessible at http://ceprofs.tamu.edu/dford/SDPMbibliography070606.pdf and at http://www.interscience.wiley.com/jpages/0883-7066/suppmat/sdr.377.html.

Research and application in project dynamics has focused on understanding the drivers of cost and schedule overrun in particular situations, and then on developing actions that either avoid or minimize the overruns, or on obtaining compensation for the additional costs. While there is some overlap, the research and applications address one of four general categories of project management: (1) post-mortem assessments for disputes and learning; (2) project estimating and risk assessment; (3) change management, risk management, and project control; and (4) management training and education. We briefly describe the general area of application and the role system dynamics models have played, then illustrate with specific applications that have been published, and conclude with policy insights and directions for future research in each area.

Post-mortem assessments for disputes and learning

Many applications of system dynamics involve a post-project assessment of what happened—how did the project deviate from the original plan, and why? The most numerous of these involve disputes between the owner/financer of the project and the contractor/executer of the project. For example, a dispute between Ingalls Shipbuilding (contractor) and the U.S. Navy (owner) is described in Cooper (1980). But post-mortem assessments also involve attempts to learn from one project to the next within an organization.

Disputes Projects involving an owner and a contractor often engender disputes over interpretations of the specific requirements of the contract, changes requested by the owner, or external events such as strikes. In many cases, such disputes involve claims of "delay and disruption"—in our terminology, ripple and knock-on effects—that might result from these specific problems. In these cases, when trying to understand why project performance differed from

Published online in Wiley InterScience
(www.interscience.wiley.com) DOI: 10.1002/sdr

the plan, all changes from conditions assumed in the original plan must be specified, regardless of responsibility.

Client-responsible changes often include: increasing project scope or altering the original design requirements; taking longer than specified in the contract to review and approve design drawings, to provide information about equipment provided by the client or another contractor, or to provide key components or test equipment. Contractor responsible changes might include failure to obtain resources in a timely fashion, perhaps as a result of delays in other projects. These changes and delays from the contracted scope of work are often referred to as the "direct impact" of client (or contractor) actions on the project.

These direct impacts often trigger controlling feedbacks and resultant "ripple and knock-on effects" caused by the positive feedbacks described earlier—"delay and disruption" in the jargon of project management disputes. In its application to such disputes, the system dynamics model is used to quantify and explain the impact of these direct changes to the project on its final cost, including the ripple effects. The model can be set up to represent the project as it actually occurred, including the direct impacts, and calibrated to the actual performance of the project. Then client-responsible direct impacts are removed and the model re-simulated to determine what would have happened without the disruptive actions of the client. The difference between the historical and "would have" simulations is the full cost of the client actions, including ripple and knock-on effects. Because of their broad boundary, including representation of the many positive feedbacks created by the indirect impacts of the contractor's and customer's control actions, system dynamics is ideally suited to determine the magnitude of these ripple effects and explain their origins. It can also apportion costs to the client, to other parties, and to the contractor through simulations removing different groups of direct impacts.

A significant number of applications of project dynamics have been for delay and disruption disputes. PRA has done more than 45 such projects (Stephens *et al.*, 2005). All have been settled out of court on favorable terms to the contractor (the usual PRA client), with the typical award averaging 50% more than with traditional dispute resolution approaches, supporting the power of system dynamics to add value to these investigations. PRA's dispute work is described by Cooper (1980) and Weil and Etherton (1990), and summarized by Sterman (2000). The Strathclyde Group have also successfully used system dynamics to support six delay and disruption claims ranging in value from U.S. $50 million to $350 million, as cited in Howick and Eden (2001).

PROJECT-TO-PROJECT LEARNING Another important use of system dynamics modeling has been in post-project evaluation. In many ways, the process of post-project evaluation is similar to a delay and disruption analysis: the model is set up to represent what actually happened on the project, including the direct impact of any changes to the project from the original plan. These changes can include externally caused changes as noted above, as well as internally

Published online in Wiley InterScience
(www.interscience.wiley.com) DOI: 10.1002/sdr

generated changes such as delays in obtaining staff or other resources, the implementation of new processes or procedures, and changes in management policy. Then the direct impacts of these changes are removed as inputs to the simulation, one at a time, to identify their contribution to any project overrun. In this way, project managers can learn which changes had the greatest impact on the project, and thereby identify risks that should be addressed in future projects. In addition, to the extent that any new management initiatives were introduced on a project, project managers can test their impact on performance, and thus decide if they should be implemented on other projects.

For example, Lyneis *et al.* (2001) and Cooper *et al.* (2002) describe the use of a model by PRA to assess the lessons learned from a comparison of three command and control system projects at Hughes Aircraft Company. The effort identified the major external and internal drivers of differences in project costs, and thereby identified management initiatives to be adopted on future projects. Abdel-Hamid and Madnick (1991) and Abdel-Hamid (1993a, 1993b) applied their software project dynamics model to five organizations during model development, and five others after model completion (two by the authors and three by others). These assessments were used primarily to determine what happened on the projects, and what would have happened had different estimating methods been used, or other staffing/schedule decisions been taken.

Post-mortem assessments: insights and future work A significant number of system dynamics project models have involved post-mortem assessments. Especially on disputes, the payoff for demonstrating delay and disruption is high, the costs of modeling relatively low, and the data relatively complete and generally available—all conditions which favor effective use of system dynamics. Post-mortem assessments for learning have been less numerous, but for project-based organizations can be just as valuable. System dynamics is a scientific method for assessing what went right and what went wrong on a project, and therefore provides the raw materials for many of the other uses of system dynamics discussed below (estimating, risk assessment, control).

Perhaps, then, the greatest need in this arena is for greater documentation and discussion of the process of using models for such assessments, and for published success stories and lessons learned that can serve as exemplars for future work. Greater discussion of the process is warranted because there seems to be some divergence in approach. For example, PRA starts with calibrating the model to what actually happened on the project and removes direct impacts, while the Strathclyde group generally starts with a calibration to the plan and adds direct impacts. Are both approaches acceptable? When is each approach preferable? There may also be some methodological (and perhaps legal) issues about the proper way to conduct "would-have" analyses, i.e., in what order the direct impacts should be removed to assess delay and disruption. PRA usually remove the direct impacts of client actions one at a time in reverse chronological order. Although this sequence makes intuitive

Published online in Wiley InterScience
(www.interscience.wiley.com) DOI: 10.1002/sdr

sense, comparisons to other possible sequences have not been published. Another process issue is how uncertainty in parameter inputs (and model structure) should be considered. Graham *et al.* (2002a, 2002b) discuss the use of Monte Carlo simulations to develop confidence bands for key metrics such as the magnitude of delay and disruption.

Given the continued poor performance of projects, the failure of organizations to devote significant effort to project-to-project learning seems inexplicable. Documentation of success stories, both for disputes and for learning, can demonstrate the value of using system dynamics for project management. A common progression within a project organization is (1) first use on a dispute, (2) followed by use for estimating and management of a new project, and (3) finally use on additional projects and project-to-project learning. Documentation of success stories and process can facilitate this progression.

Project estimating and risk assessment

Strong anecdotal evidence suggests that, in addition to changes to the plan, another common trigger for adverse project dynamics is underestimating work scope or under-budgeting for the estimated work scope. While post-project assessments are essential for understanding what happened, their greatest value may be in improving project estimating and risk assessment—how can we develop project budgets and plans that are more realistic and robust?

PROJECT ESTIMATING Abdel-Hamid and Madnick (1991) and Abdel-Hamid (1993b) discuss the use of system dynamics models in conjunction with more traditional estimation approaches (such as COCOMO for software) to develop project estimates of effort and time requirements. Abdel-Hamid argues that this can and should be done at three stages during a project: (1) upfront, to adjust traditional estimates based on known or expected deviations (risks) from typical projects; (2) during the project, to determine the degree of any project underestimation earlier than would typically occur; and (3) after the project, to assess what the project should have cost had other decisions been taken, including better initial estimates. This last assessment is critical, as he demonstrates how project estimates can affect the final schedule and cost of a project—projects that are underestimated end up costing more because of the adverse ripple effect dynamics incurred once the underestimate is discovered; projects that are overestimated also end up costing more than they otherwise would because of the tendency to slack off and/or “gold-plate” when there is insufficient schedule pressure. He also shows how similar dynamics can lock in project-to-project underperformance of productivity-enhancing tools (Abdel Hamid, 1996), and demonstrates this phenomenon in a series of controlled experiments using a gaming version of the model (Sengupta and Abdel-Hamid, 1996). Adjusting estimates based on a system dynamics model can help reduce or eliminate these dynamics.

Published online in Wiley InterScience
(www.interscience.wiley.com) DOI: 10.1002/sdr

Risk assessment All project plans make assumptions about uncontrollable factors that might affect a project. These include, but are not limited to, assumptions that resources and/or skills can be obtained as planned, delays in receiving information and/or materials from other projects and vendors will not be excessive, technical uncertainties will be resolved in a timely manner, and new tools and organization structures will work as planned. Risk assessment asks the question: "If certain assumptions in the baseline plan are not met, what would be the impact on project performance?" System dynamics has been used for risk assessment in two ways: (1) post-project evaluations determine the magnitude of changes that actually occurred on projects as a guide for what might occur on future projects; and (2) pre-project simulations test the consequences of similar risks for the current project. For example, PRA did a post-mortem analysis of 11 development projects at a major automotive company. The analysis identified and quantified five continuing sources of risk (causes of overrun) for such development projects: (1) late information and/or changes; (2) resource availability (slow ramp-up, lower peak, forced ramp-down to meet budget, inadequate skills mix); (3) new processes, missing enablers, or new materials; (4) organization and/or geographic changes; and (5) aggressive program assumptions (stretch objectives causing compressed timing, inadequate budget, or lean allowance for prototypes). They also suggested actions to mitigate these risks. While this work, and similar work by others that the authors are aware of, has proved to be of significant benefit to the company, it has not been published.

Project estimating and risk assessment: insights and future work Project plans sow the seeds for project success or failure. Why are managers biased toward continued underestimates of true costs in the face of continued evidence of project overruns, and how can they be convinced that the first step to avoiding adverse project dynamics is to bid and plan the project correctly?

In all likelihood, managers' continued underestimation of project budgets partly reflects the inherent difficulties in estimating the scope of work on a complex development, and partly management's (and staff's) tendency to underestimate the effort required. This underestimation, at least on management's part, likely reflects some combination of (1) fear that the project will not be accepted or continued if the budget estimate is too high; (2) desire to put pressure on staff to avoid the "gold-plating" and slacking-off phenomena noted above; (3) failure to adequately budget for the hours and time needed to perform even normal rework, or for the productivity and quality costs of planned concurrence; and (4) the belief that aggressive stretch objectives maximize performance while trying to achieve an unrealistic plan does not have any adverse consequences (underestimation of ripple effects). Some managers apparently believe "Sure, the project is underestimated, but what's the worst that can happen? We'll add resources or schedule and end up with what a reasonable plan would have produced. And maybe we'll be able to pressure

Published online in Wiley InterScience
(www.interscience.wiley.com) DOI: 10.1002/sdr

the staff to do better and actually reduce our costs." Project planners find it seductively easy to ignore the adverse dynamics created when a project falls behind and actions are taken to bring it back on schedule. One significant contribution that system dynamics has, and can continue to make, is to convincingly persuade management that trying to achieve an overly aggressive plan actually makes the performance of the project worse.

Another valuable area for future research would be in explicitly identifying the reasons why project budgets are continually underestimated. Repenning and Sterman (2001, 2002) have done research that might provide one answer. They argue that the solution managers choose for a problem, for example, poor performance, depends on their attribution of the cause. If they believe the problem results from inadequate processes, then they will put effort into process improvement; if they believe the problem results from inadequate worker effort ("slacking-off"), then they will put effort into increasing the quantity of work. Repenning and Sterman document numerous reasons why managers tend to favor the latter (inadequate worker effort) over the former (inadequate processes). They argue that

> Managers' tendency to attribute performance shortfalls to problems with the workforce rather than the production system is reinforced by the so-called fundamental attribution error, or dispositional bias. Attributing a problem or behavior to individuals rather than the systems in which they are embedded is a pervasive and robust phenomenon. (Repenning and Sterman, 2002: p. 285)

They further show how the tendency to blame workers rather than processes can become self-confirming. These results may have implications for project scoping and estimation. It is easier for managers to believe that project problems have been caused by individual lack of effort, rather than by the systemic ripple and knock-on effects that cause low productivity and increase errors. In this case, the tendency would be to underestimate resource requirements or schedule to exert pressure on the workforce, instead of changing the policies and processes in which the workforce operates. Further research into persistent underestimation of project work and resource requirements is needed.

Once the reasons for underestimation are identified, the question of how better estimates should be determined remains. As discussed above, post-mortem analysis of prior projects can provide a baseline model estimate of the planned project. However, even with these estimates, the true scope and cost of the next project cannot be known with certainty in advance. Therefore, is it better to err on the high or the low side? Under what conditions? What degree of slack or buffers are appropriate, and who on project teams should know about and have control over such buffers? Initial work (Ford, 2002) should be expanded.

Finally, uncertainty is omnipresent in projects, causing risks to meeting objectives. Although, as described, risk has been addressed by system dynamics

Published online in Wiley InterScience
(www.interscience.wiley.com) DOI: 10.1002/sdr

project modelers, the full power of the strategic perspective possible with system dynamics has not been used to design or analyze risk strategies that apply several risk management tools or integrate with existing risk management theory. Robustness—the ability to deliver good performance under uncertainty—is a holy grail of project management that system dynamics suggests is attainable with good planning and the appropriate use of adaptive control. But robustness is difficult to measure and harder to design. Developing tools to assess and improve project robustness is a rich opportunity for system dynamics. How can robustness be measured and used as a project planning and management performance measure? Which project components and policies affect robustness most? What processes and policies improve robustness? Initial efforts such as Taylor and Ford (2006) should be extended and expanded.

Change management, risk management, and project control

Even with improved planning, projects will rarely go exactly as planned. When problems occur, how should management best respond? To what extent can additional budget be obtained ("change management")? How can risks be mitigated ("risk management")? What mix of adding resources (e.g., by hiring, overtime, work intensity), changing the schedule (both final and interim milestones), reducing scope, cutting activities such as QA, and so on will provide the most satisfactory outcome? A system dynamics model can provide valuable input into such decisions by taking into consideration feedback in projects, especially the adverse ripple effects of management actions.

CHANGE MANAGEMENT When customers make changes to projects, the original plan almost always becomes infeasible. Change management entails pricing and mitigating proposed changes as they occur on an ongoing project (rather than waiting for disputes to occur after the project ends). Cooper and Reichelt (2004) demonstrate that the full cost of changes, including ripple effects, increases nonlinearly with the cumulative size of all changes, and as the changes occur later in the project. Eden *et al.* (2000) call these "Portfolio Effects", where combinations of changes produce impacts greater than the sum of the individual impacts alone. Many clients and project managers overlook these ripple effects when requesting, pricing, and accepting changes—they typically price the changes at the estimated direct costs of the added scope. As a result, the project overruns the schedule and/or budget, and disputes are likely to arise.

Examples of applications of system dynamics to change management include the following:

- Williams (1999) describes the use of their model to assess optimal schedule extensions when changes are introduced on a project.
- Howick and Eden (2001) examine the consequences of attempting to compress a project's schedule, at the request of the client, after the project

Published online in Wiley InterScience
(www.interscience.wiley.com) DOI: 10.1002/sdr

has started, and demonstrate that too often contractors ask for insufficient compensation because they ignore the ripple effects.

- Fluor Corporation proactively uses project models (dozens to date) to forecast and mitigate change impacts, including quantifying the changes' effects, diagnosing the causes, and planning and testing mitigating actions to reduce project costs. Fluor reports that their clients welcome their use of these models, appreciating the foresight that helps avoid project cost surprises and minimize capital expenditures. This has required extensive investment in management education, training well over 1000 managers throughout the company's offices internationally.

Risk management System dynamics project models have also been applied to investigate risk management as an aspect of project management distinct from project control. While system dynamicists have used project models to investigate the effectiveness and use of specific risk management tools or strategies, they also develop insight by focusing on risk management approaches instead of specific policies. Managerial flexibility is an example. Ford and others use system dynamics to operationalize real options theory in projects for risk management. Case studies (Ford and Ceylan, 2002; Alessandri *et al.*, 2004; Johnson *et al.*, 2006) and comparisons with other approaches (Cao *et al.*, 2006) establish a basis for the feedback role in managerial real options. System dynamics model structures specify real options decision making and test option valuation theory (Bhargav and Ford, 2006). Ford and Sobek (2005) applied this approach to a product development project to more fully describe Toyota's unique product development approach to managing design risk and to partially explain Toyota's industry-leading performance. Adopting a similar approach, Johnson *et al.* (2006) use system dynamics to model and value flexibility in equipment delivery strategies in a large petrochemical project.

Managerial mental models about risk provide a second example. Project managers simultaneously seek project structures and policies that maximize project performance yet perform well when faced with a range of uncertainties that reflect risks. System dynamics research has shown that managers who tailor polices for specific project assumptions can outperform those that manage for a wide range of conditions, *if those project assumptions materialize*. But if conditions deviate from those assumed by management, tailored policies generate much worse performance. Several researchers in different contexts have identified this fundamental trade-off between project robustness, the ability to perform well across a range of uncertain conditions, and performance under known specific conditions. Repenning (2000) first identified this trade-off using system dynamics. Ford (2002) found a similar trade-off by modeling practitioner mental models of budget contingency management. Park and Pena-Mora (2004) propose and test a strategy of schedule buffer allocation that includes overlapping to allow more time for quality assurance in downstream activities. They found that sizing and locating schedule buffers

Published online in Wiley InterScience
(www.interscience.wiley.com) DOI: 10.1002/sdr

can improve project performance by reducing the impacts of changes for a particular type of risk.

PROJECT CONTROL Projects rarely go as planned. When problems occur, how should management best respond? Building from the research and applications in the field, and especially Graham (2000), Cooper (1994), and Smith *et al.* (1993), we summarize the key project control lessons that come from understanding project dynamics into two categories: managing the rework cycle, and minimizing ripple and knock-on effects.

The rework cycle is central to many adverse project dynamics. If the rework cycle is recognized, management can take actions to minimize its consequences. Specifically, system dynamics project models have been used to identify how managers can:

- Improve quality and reduce errors, even if those efforts reduce productivity—doing work fewer times, even at lower productivity, is generally beneficial. One approach is to slow down and do work right the first time (i.e., reduce work intensity), even if this might cause some "slacking off". Another approach uses integrated product teams, which improve quality and rework discovery at the expense of reduced productivity from greater communications overheads. Graham (2000) argues that to be effective these teams should include customers, and all functions, and that the people on the teams should have the knowledge and authority to make decisions that will improve the end product.
- Recognize the existence of undiscovered rework and avoid its consequences, primarily the "errors create more errors" dynamic. Undiscovered rework can be reduced, for example, by prioritizing rework detection and correction over starting new work. Park and Pena-Mora (2004) investigate how to have staff spend time (re)checking before starting new work to operationalize this strategy (see also Lyneis *et al.*, 2001). Early testing to discover problems rather than testing to pass tests can also reduce rework cycle consequences.
- Avoid the tendency to start downstream work too early and thereby increase unplanned concurrence, reallocate "excess" staff that will be needed later when the rework is discovered, or both. Another consequence of undiscovered rework is that a project is likely to be further behind than typical reporting systems indicate. This can lure managers into earlier downstream phase initiation or staff reductions that generate knock-on effects.
- Use a formal model to help implement improved policies. Even if recognized and designed well, the project control actions above are often difficult to implement because implementation initiates a "worse before better" behavior mode. By effectively demonstrating any worse-before-better dynamics and the eventual benefits of implementation, a formal model can give managers the courage to stick with implementation. For example, allocating more resources to QA reduces "perceived progress" while actually

Published online in Wiley InterScience
(www.interscience.wiley.com) DOI: 10.1002/sdr

increasing "real progress". Such actions may be difficult to stick with unless managers have confidence that the "better" will occur after the "worse".

In addition to improving behavior through management of the rework cycle, managers can significantly improve project performance through efforts to manage ripple and knock-on effects—how managers respond when the existence of an infeasible initial plan is discovered, or when changes or other risks materialize (thereby making the plan infeasible), has a significant impact on project dynamics. Two types of project control actions are available:

- Ease performance targets, such as by slipping the completion or milestone deadlines, increasing the budget, reducing the scope, or accepting a higher fraction of flaws in the final product. These actions reduce ripple effects by reducing the need to change project management and progress. Easing targets would seem to be more attractive when it is difficult or expensive to change the performance of the project.
- Increase effective resources, such as by adding staff, working overtime, or increasing work intensity or by using staff more efficiently. These actions can initiate ripple and knock-on effects.

CHANGE MANAGEMENT, RISK MANAGEMENT, AND PROJECT CONTROL: INSIGHTS AND FUTURE WORK While the literature stresses the importance of minimizing the rework cycle and avoiding positive feedbacks that operate as vicious cycles, it is often not specific. For example, Smith *et al.* (1993) offer the following policy advice: (1) extra time spent during requirements and design result in a higher quality project at lower cost; (2) increasing personnel on a project is usually counterproductive—implement a project with a fixed number of staff from the beginning; (3) sustained overtime does not increase productivity in the long run; (4) a moderate amount of schedule pressure is optimal; and (5) the use of "experts" can significantly improve project performance. While these seem logical, are they true on all projects, and what specific actions should be taken? What are "sustained" or "moderate" levels that should be avoided or followed? Beyond Graham's (2000) assertion that "managers should avoid use of sustained overtime (longer than 3 months)", there is little quantitative guidance or support for the general advice offered.

The lack of published advice may reflect the conflict between seeking recommendations that are both widely applicable and have rigor based on specific projects and issues. But the lack of published advice may also reflect the fact that research in this area is lacking. While researchers and practitioners have examined specific situations, few have attempted to generalize recommendations. This is an important area for future research. What project conditions should managers use as the basis for project control decisions? What should managers do when a project is forecast to fall short of performance targets? What combination, order, and duration of easing targets and

Published online in Wiley InterScience
(www.interscience.wiley.com) DOI: 10.1002/sdr

increasing effective resources bring the project closest to its performance goals? The best approach to getting the project back on track is not obvious. From a dynamic systems perspective, this is particularly difficult to ascertain because the strengths of the feedback loops differ across projects and are dynamic during projects. What heuristics for managing project dynamics improve performance? What qualitative and quantitative models help develop, teach, and train about these heuristics? Each project is different, so the literature cannot offer specific advice such as "use *x*% overtime while hiring *y*% more staff". However, more work is needed on how managers can use the insights from the system dynamics literature or from a project-specific models to develop such guidelines for their particular projects. Ford *et al.* (2007) have initiated one research project along these lines.

Sengupta and Abdel-Hamid (1993) use their model in a gaming format to show how system dynamics models can be used to generalize about project management. They showed that student managers perform best when given cognitive feedback (e.g., information on fraction of workforce experienced, productivity, communication overhead), worse when given feed-forward feedback (e.g., heuristics for hiring), and worst when given outcome feedback (e.g., estimated progress, hours spent). That is, students do worst with typical management information, but can improve with heuristics and information targeted at the cause of adverse dynamics. Similar questions apply to other project controls. For example, Lee *et al.* (2007) investigate the interaction of resource allocation delays and different amounts of control imposed by managers. Their results suggest general but counter-intuitive project control recommendations, such as to exert less control to decrease project durations. Assuming these and other general project control lessons prove effective, how can project mangers be convinced to adopt them?

While at least some work has been done examining actions to bring a project back on schedule, little work has been done on the consequences of closing the performance–target gap by changing project deadlines. More generally, the secondary consequences of adjusting project targets has not been as well studied in system dynamics as the secondary consequences of adjusting effort and resources. PRA has usually modeled slipping schedule as well as adding resources; however, they rarely examine secondary impacts of such slips on performance of the product in the market. Additional work is needed in understanding the secondary consequences of controls to achieve quality and budget targets, and of actions to adjust the targets themselves. However, adequately representing the secondary consequences of target adjustments will expand the boundary of project models to include interactions with the market (impact of delays in reaching the market, project and product cost, and product features on market success) and other parts of the firm (impact of resource usage on other projects).

Multiple performance measures that vary in importance across project participants and time are a hallmark of projects. While schedule targets are often the top priority on projects, cost, delivered quality, and/or scope can also be critical. Relatively little analysis of the dynamics created by attempts to meet these targets, along the lines shown in Figure 4, has been done. How does management of various performance measures differ? The management of different performance measures is complicated by the interdependence of performance measures (e.g., longer projects can cost more because of "marching army" fixed costs). System dynamics project models can demonstrate traditional performance trade-offs (e.g., between duration and cost) inherent in project management (e.g., the use of overtime). But to add value system dynamics needs to contribute deeper insights about multiple performance measures. The work to date has demonstrated many ways in which managers can *fail*. How can managers proactively *succeed* when faced with conflicting performance measures? Most system dynamics models of projects assume a single set of performance priorities. But project practice typically includes important differences in performance priorities across the project team. What is "best" often differs among project participants (e.g., owner, designer, builder). For example, to an owner the best solution to a late project may be increased builder's staff. Although that may increase costs and reduce the builder's profit, the owner can retain the planned benefits. In contrast, the best solution from the builder's perspective in the same project may be to slip the completion deadline. Although this may delay and therefore reduce the owner benefits, it can minimize the builder's costs and retain the builder's profit. How can the competition among project participants with different targets and priorities be modeled and improved? System dynamics models can be developed to address the relative winners and losers within project teams when projects are managed with certain approaches and policies.

Management training and education

Project models are often used to teach system dynamics. System dynamics project models have been used with both practicing project managers and students in formal educational settings. The familiarity of projects and their frequent poor performance are of great interest to managers (as suggested by the popularity of the comic strip "Dilbert"), making projects popular units in system dynamics courses. These applications typically are limited to a focused case study or building relatively simple models to illustrate the system dynamics method with a few of the structures and dynamics described above. The ability of system dynamics to clearly and richly explain how project structures and behavior interact also makes it effective for teaching project management. Most uses for this purpose in schools are graduate-level courses that depend on a fundamental understanding of project management from undergraduate coursework or practical experience.

A number of the project models have been converted into gaming simulators for use in management training, for example at BP, Bath, Ford, Hughes/Raytheon,

Published online in Wiley InterScience
(www.interscience.wiley.com) DOI: 10.1002/sdr

and the World Bank. Trainees manage a simulated project, making typical decisions such as hiring, use of overtime, exerting schedule pressure, allocation of staff to various activities, when to start downstream phases of work, and so on. After such a simulation, the reasons for simulated project performance are analyzed, usually using diagrams such as those developed above, and diagnostic output from the models. Trainees are then often allowed to manage the simulated project again. The Strathclyde group (Howick and Eden, 2001) have converted their claim models into a generic project model they have used for management training. Abdel-Hamid has developed a simulator version of his software dynamics model, but seems to have used it primarily to conduct experiments on how managers make decisions (as discussed earlier). In addition, Barlas and Bayraktutar (1992), Repenning (MacInnis, 2004), and others have developed gaming simulators for use with students and managers.

As an aid in teaching system dynamics, project models have and will continue to play a valuable role. However, project models have only begun to fulfill their potential in teaching project management and project control. The experience of one of the authors in teaching project management indicates that two of the largest challenges for students, especially those with little or no industry experience, are to understand the complexity of projects that the many causal relations create, and to appreciate at a visceral level the challenges of project management. System dynamics project models can help effectively address both of these challenges by being the basis for project management flight simulators. Beyond using project models as black-box proxies for actual projects, the transparent nature of the models developed for project management training and education can be leveraged to demonstrate how a scientific investigation of the feedback structure of projects can improve managerial understanding, policies, and performance. Existing models can be the basis for such teaching tools. However, a good project model is only one part of an effective management flight simulator—supporting diagnostic and educational materials are also needed. Developing a useful system dynamics project model-based teaching tool for widespread project management training and education is a relatively large endeavor, but could make major advances in project management education and the application of system dynamics. Existing and new models and tools should be developed for widespread use.

Discussion and conclusions

Measured both in terms of academic research and real-world applications, the use of system dynamics to understand and improve project management has been a great success. This success includes significant advancement in system dynamics theory, new and improved model structures, value added in practice, and growth of the system dynamics field. We summarize what has been accomplished and what remains to be done in three categories: (1) theory

Published online in Wiley InterScience
(www.interscience.wiley.com) DOI: 10.1002/sdr

development; (2) guidance in improving project management and education; and (3) applications.

Theory development

The primary causes of project dynamics—project features, the rework cycle, project controls, and ripple and knock-on effects—are in our view nearly completely and adequately represented in existing project models, especially as they pertain to controls to achieve schedule performance. While no doubt special structures are needed to represent particular types of projects, these special structures are at the fine-tuning level and do not represent fundamental additions to the drivers of project dynamics. One important area of research may be in identifying how the strengths of the various ripple and knock-on effects may differ by phase of work (e.g., design vs. construction) or by type of project (e.g., software vs. hardware), or as a result of changes in process and tools (e.g., the adoption of CAD systems), and how different strengths may alter any policy heuristics. Ideally, empirical studies would be conducted to identify differences, and simulation analyses to translate those differences into policy guidance. Additional work is needed in understanding the secondary consequences of controls to achieve quality and budget targets, and of actions to adjust the targets themselves. However, adequately representing the secondary consequences of target adjustments will expand the boundary of project models to include interactions with the market and other parts of the firm.

Guidance in improving project management and education

While our understanding of what causes project dynamics is fairly complete, much work remains in translating that theory into improved project management and education. In which directions should system dynamics project modeling develop? What issues and topics are the most fertile for continuing to improve our understanding of project dynamics? Our review suggests valuable issues and questions in several areas. We offer these not only as suggestions for future work but also to initiate and catalyze a discourse among system dynamics project modelers about future directions.

- *Post-mortem assessments:* While some research on the approach to conducting simulation experiments is warranted, as discussed above, perhaps the greatest need is for published success stories and lessons learned that can serve as exemplars of success and guidance for future work.
- *Project estimating and risk assessment:* The persistent and large underestimation of projects may provide the best opportunity for system dynamics to improve projects. Why do managers continue to underestimate projects in the face of continued evidence of project overruns? How much do they underestimate ripple effects, and why? What can be done to improve these

Published online in Wiley InterScience
(www.interscience.wiley.com) DOI: 10.1002/sdr

practices? Case and empirical studies of projects would seem to be the best approach to these questions. If we could get managers to attempt to make more accurate estimates, how should this best be done? While we may have prior project experience, and even models, to develop base estimates, the true scope and cost of the next project cannot be known in advance. Therefore, is it better to err on the high or the low side? Under what conditions? What degree of slack, buffers, and budget contingencies are appropriate, and who on project teams should know about and have control over what buffers? Here, simulation analyses in concert with real applications would seem to be the best approach.

- *Change management, risk management, and project controls:* What project conditions should managers use as the basis for project control decisions? What should managers do when a project is forecast to fail to meet performance targets? In what amounts, durations, combinations, and orders of application should individual project controls such as hiring or overtime be applied to improve performance as much as is possible? What heuristics for managing project dynamics improve performance? What qualitative and quantitative models help develop, teach, and train about these heuristics? These are all questions that should rigorously be studied using simulation models. Similar questions apply to other project controls. While schedule targets are often the top priority on projects, cost, delivered quality, and/or scope can be critical. Relatively little analysis of the dynamics created by attempts to meet these targets, along the lines shown in Figure 4, has been done. How does management of various performance measures differ? Project performance measures are interdependent. How do interactions among performance measures affect project success?
- *Management training and education:* While project-management flight simulators exist, and project dynamics ideas are taught in a few courses, the use of system dynamics concepts to help managers understand project behavior and improve management is limited. Perhaps the biggest need is teaching materials that can be used by both system dynamicists and by others, at a range of depth and duration, to first enhance management awareness, and then to improve the skills of those that might use system dynamics models in support of project management.

Applications

Even though the application of system dynamics to project management has been significant relative to other system dynamics work, system dynamics is used on a relatively small percentage of projects. In addition, system dynamics modeling is typically applied to individual projects, not to all projects across an organization. Why? And how can we increase its use in a field which clearly needs better management? Based on our survey, three things might improve diffusion: (1) publication of more success stories, especially in the project

Published online in Wiley InterScience
(www.interscience.wiley.com) DOI: 10.1002/sdr

management literature; (2) making system dynamics models easier and less costly to develop; and (3) better integration of system dynamics models with traditional project management tools.

In spite of the hundreds of applications of system dynamics to project management, relatively few papers have been written and even fewer in the journals widely read by practicing project managers. Moreover, few project management courses include system dynamics as a component. As in other areas of system dynamics, applications to project management are time-intensive and as a result costly. In addition to greater training, development of "packaged" project models, components, and tools to support model setup and refinement (e.g., integration with databases and calibration software) will help to reduce the effort and cost of applications.

Finally, some have suggested that integration of system dynamics with more traditional tools can help spread its use. System dynamics modeling is more strategic/tactical in nature than more traditional operational project management tools such as work breakdown structures, critical path modeling, and component cost estimating (Rodrigues and Bowers, 1996). The ability of system dynamics modeling to complement traditional project management tools by adding a strategic and tactical perspective can add value in combination with traditional tools, a common experience in system dynamics project model applications. Rodrigues (2000) and Rodrigues and Bowers (1996) describe a methodology to integrate the use of system dynamics within the established project management processes. In Rodrigues (2001), this is further extended to integrate the use of system dynamics modeling within the PMBOK risk management process, providing a useful framework for managing project risk dynamics. Pfahl and Lebsanft (1999) recommend integration of system dynamics modeling with descriptive process modeling and goal-oriented measurement. Park and Pena-Mora (Park, 2001; Park and Pena-Mora, 2003, 2004) propose and apply an integration of dynamic schedule buffering using a system dynamics model with critical path modeling. Williams (2002) devotes a chapter to the topic, discussing how system dynamics can improve traditional models, how traditional tools can be used to create system dynamics models, and how system dynamics and traditional models can be used to inform one another. Integrating system dynamics project models and other project management tools might improve application, and therefore warrants further research on integration techniques, trial applications, and dissemination of successful approaches.

The application of system dynamics to project management demonstrates the power of the methodology to build theory and improve practice. But greater impact is possible. Expansion of models and applications can be facilitated with a broader dissemination of existing and future work. The nature and diversity of projects provide ample opportunities for continued development and application. We welcome a dialog with academics and practitioners to further the work.

Published online in Wiley InterScience
(www.interscience.wiley.com) DOI: 10.1002/sdr

Notes

1. Pugh–Roberts Associates is now a part of the PA Consulting Group.
2. Versions of the PRA rework cycle with equations are given in Richardson and Pugh (1981) and Lyneis (2003).

References

Abdel-Hamid TK. 1984. The dynamics of software development project management: an integrative system dynamics perspective. PhD thesis, MIT Sloan School of Management, Cambridge, MA.

——. 1993a. A multiproject perspective of single-project dynamics. *Journal of Systems Software* **22**(3): 151–165.

——. 1993b. Adapting, correcting, and perfecting software estimates: a maintenance metaphor. *Computer* March: 20–29.

——. 1996. The slippery path to productivity improvement. *IEEE Software* July: 43–52.

Abdel-Hamid TK, Madnick SE. 1991. *Software Project Dynamics*: *An Integrated Approach*. Prentice-Hall: Englewood Cliffs, NJ.

Ackermann F, Eden C, Williams T. 1997. Modelling for litigation: mixing qualitative and quantitative approaches. *Interfaces* **27**(2): 48–65.

Alessandri T, Ford D, Lander D, Leggio K, Taylor M. 2004. Managing risk and uncertainty in complex capital projects, *Quarterly Review of Economics and Finance* **44**(5): 751–767.

Barlas Y, Bayraktutar I. 1992. An interactive simulation game for software project management (Softsim). In *Proceedings of the 1992 International System Dynamics Conference*, Utrecht, The Netherlands.

Bhargav S, Ford DN. 2006. Project management quality and the value of flexible strategies, *Engineering, Construction and Architectural Management* **13**(3): 275–289.

Cao Q, Ford D, Leggio K. 2006. The application of real options to the R&D outsourcing decision. In *Outsourcing Management Information Systems*, Schniederjans M, Schniederjans A, Schniederjans D (eds). Idea Group Publishing: London.

Cooper KG. 1980. Naval ship production: a claim settled and a framework built. *Interfaces* **10**(6): 20–36.

——. 1993. The rework cycle (a series of 3 articles): why projects are mismanaged; how it really works . . . and reworks . . . ; benchmarks for the project manager. *PMNETwork* February (for first two articles); *Project Management Journal* March (for third article).

——. 1994. The $2,000 hour: how managers influence project performance through the rework cycle. *Project Management Journal* **25**(1).

Cooper KG, Reichelt KS. 2004. Project changes: sources, impacts, mitigation, pricing, litigation, and excellence. In *The Wiley Guide to Managing Projects*, Morris PWG, Pinto JK (eds). Wiley: Hoboken, NJ; pp. 743–772.

Cooper KG, Lyneis JM, Byrant BJ. 2002. Learning to learn, from past to future. *International Journal of Project Management* **20**: 213–219.

Eden CE, Williams TM, Ackermann FA. 1998. Dismantling the learning curve: the role

Published online in Wiley InterScience (www.interscience.wiley.com) DOI: 10.1002/sdr

of disruptions on the planning of development projects. *International Journal of Project Management* **16**(3): 131–138.

Eden CE, Williams TM, Ackermann FA, Howick S. 2000. On the nature of disruption and delay (D&D) in major projects. *Journal of the Operational Research Society* **51**(3): 291–300.

Ford DN. 1995. The dynamics of project management: an investigation of the impacts of project process and coordination on performance. PhD thesis, MIT Sloan School of Management, Cambridge, MA.

——. 2002. Achieving multiple project objectives through contingency management. *ASCE Journal of Construction Engineering and Management* **128**(1): 30–39.

Ford DN, Anderson S, Damron A, de Las Casas R, Gokmen N, Kuennen S. 2004. Managing constructability reviews to reduce highway project durations. *ASCE Journal of Construction Engineering and Management* **130**(1): 33–42.

Ford DN, Ceylan K. 2002. Using options to manage dynamic uncertainty in acquisition projects. *Acquisition Review Quarterly* **9**(4): 243–258.

Ford DN, Lyneis JM, Taylor T. 2007. Project controls to minimize cost and schedule overruns: a model, research agenda, and initial results. In *Proceedings of the 2007 International System Dynamics Conference*, Boston, MA.

Ford DN, Sobek D. 2005. Modeling real options to switch among alternatives in product development. *IEEE Transactions on Engineering Management* **52**(2): 1–11.

Ford DN, Sterman JD. 1998a. Dynamic modeling of product development processes. *System Dynamics Review* **14**(1): 31–68.

——, ——. 1998b. Expert knowledge elicitation for improving mental and formal models. *System Dynamics Review* **14**(4): 309–340.

——, ——. 2003b. The liar's club: impacts of concealment in concurrent development projects. *Concurrent Engineering Research and Applications* **111**(3): 211–219.

Forrester JW. 1961. *Industrial Dynamics*. MIT Press: Cambridge, MA.

——. 1969. *Urban Dynamics*. MIT Press: Cambridge, MA.

Graham AK. 2000. Beyond PM101: lessons for managing large development programs. *Project Management Journal* **31**(4): 7–18.

Graham AK, Choi CY, Mullen TW. 2002a. Using fit-constrained Monte Carlo trials to quantify confidence in simulation model outcomes. In *Proceedings of the 35th Annual Hawaii Conference on Systems Sciences*, IEEE: Los Alamitos, CA.

Graham AK, Moore J, Choi CY. 2002b. How robust are conclusions from a complex calibrated model, really? A project management model benchmark using fit constrained Monte Carlo analysis. In *Proceedings of the 2002 International System Dynamics Conference*, Palermo, Italy.

Homer JB, Sterman JD, Greenwood B, Perkola M. 1993. Delivery time reduction in pulp and paper mill construction projects: a dynamic analysis of alternatives. In *Proceedings of the 1993 International System Dynamics Conference*, Cancun, Mexico.

Howick S, Eden C. 2001. The impact of disruption and delay when compressing large projects: going for incentives? *Journal of the Operational Research Society* **52**: 26–34.

Joglekar N, Ford DN. 2005. Product development resource allocation with foresight. *European Journal of Operational Research* **160**(1): 72–87.

Johnson S, Taylor T, Ford DN. 2006. Using system dynamics to extend real options use: insights from the oil & gas industry. In *Proceedings of the 2006 International System Dynamics Conference*, 23–27 July, Nijmegen, The Netherlands.

Published online in Wiley InterScience
(www.interscience.wiley.com) DOI: 10.1002/sdr

Lee Z, Ford DN, Joglekar N. 2007. Resource allocation policy design for reduced project duration: a systems modeling approach. *Systems Research and Behavioral Science* **24**: 1–15.

Levitt RE, Thomsen J, Christiansen TR, Kunz JC, Jin Y, Nass C. 1999. Simulating project work processes and organizations: toward a micro-contingency theory of organizational design. *Management Science* **45**(11): 1479–1495.

Lyneis JM. 2003. Course notes for MIT course ESD.36J: system and project management, Fall 2003. http://ocw.mit.edu/OcwWeb/Engineering-Systems-Division/ESD-36JFall-2003/CourseHome/index.htm

Lyneis JM, Cooper KG, Els SA. 2001. Strategic management of complex projects: a case study using system dynamics. *System Dynamics Review* **17**: 237–260.

MacInnis DV. 2004. Development of a system dynamics based management flight simulator for new product development. MSc thesis, System Design and Management Program, MIT.

Madachy RJ. 2002. Software process concurrence. In *Proceedings of the 2002 International System Dynamics Conference*, Palermo, Italy.

McKenna N. 2005. Executing major projects through contractors. In *Proceedings of the 2005 International System Dynamics Conference*, Boston, MA.

Park M. 2001. *Dynamic Planning and Control Methodology for Concurrent Construction Projects*. PhD dissertation, Massachusetts Institute of Technology, Cambridge, MA.

Park M, Pena-Mora F. 2003. Dynamic change management for construction: introducing the change cycle into model-based project management. *System Dynamics Review* **19**(3): 213–242.

——, ——. 2004. Reliability buffering for construction projects. *ASCE Journal of Construction Engineering and Management* **130**(5): 626–637.

Pfahl D, Lebsanft K. 1999. Integration of system dynamics modelling with descriptive process modelling and goal-oriented measurement. *Journal of Systems and Software* **46**(2–3): 135–150. 15 April 1999.

Project Management Institute. 2000. *A Guide to the Project Management Body of Knowledge* (PMBOK ® Guide). Project Management Institute: PA.

Reichelt KS. 1990. *Halter Marine: A Case Study of The Dangers of Litigation*. Masters thesis, Massachusetts Institute of Technology.

Reichelt KS, Lyneis JM. 1999. The dynamics of project performance: benchmarking the drivers of cost and schedule overrun. *European Management Journal* **17**(2): 135–150.

Repenning NP. 2000. A dynamic model of resource allocation in multi-project research and development systems. *System Dynamics Review* **16**(3): 173–212.

Repenning NP, Sterman JD. 2001. Nobody ever gets credit for fixing problems that never happened: creating and sustaining process improvement. *California Management Review* **43**(4): 64–88.

——, ——. 2002. Capability traps and self-confirming attribution errors in the dynamics of process improvement. *Administrative Science Quarterly* **47**: 265–295.

Richardson GP, Pugh AL III. 1981. *Introduction to System Dynamics Modeling with Dynamo*. MIT Press: Cambridge, MA.

Roberts EB. 1964. *The Dynamics of Research and Development*. Harper & Row: New York.

——. 1974. A simple model of R&D project dynamics. *R&D Management* **5**(1). Reprinted in Roberts EB (ed.). 1978. *Managerial Applications of System Dynamics*. Productivity Press: Cambridge, MA.

Published online in Wiley InterScience
(www.interscience.wiley.com) DOI: 10.1002/sdr

Rodrigues A. 2000. The application of system dynamics to project management: an integrated methodology (SYDPIM). PhD thesis, Department of Management Science, University of Strathclyde.

Rodrigues AG. 2001. Managing and modelling project risk dynamics: a system dynamics-based framework. Presented at the *Fourth European Project Management Conference: PMI Europe 2001*, 6–7 June, London.

Rodrigues AG, Bowers J. 1996. System dynamics in project management: a comparative analysis with traditional methods. *System Dynamics Review* **12**(2): 121–139.

Rodrigues AG, Williams TM. 1998. System dynamics in project management: assessing the impacts of client behavior on project performance. *Journal of the Operational Research Society* **49**(1): 2–15.

Sengupta K, Abdel-Hamid TK. 1993. Alternative conceptions of feedback in dynamic decision environments: an experimental investigation. *Management Science* **39**(4): 411–428.

——, ——. 1996. The impact of unreliable information on the management of software projects: a dynamic decision perspective. *IEEE Transactions on Systems, Man, and Cybernetics* **26**(2): 177–189.

Smith BJ, Nguyen N, Vidale RF. 1993. Death of a software manager: how to avoid career suicide through dynamic software process modeling. *American Programmer* **6**(5): 10–17.

Stephens CA, Graham AK, Lyneis JM. 2005. System dynamics modeling in the legal arena: meeting the challenges of expert witness admissibility. *System Dynamics Review* **21**(2): 95–122.

Sterman JD. 2000. *Business Dynamics: Systems Thinking and Modeling for a Complex World*. Irwin/McGraw Hill: Chicago, IL.

Taylor T, Ford DN. 2006. Tipping point dynamics in development projects. *System Dynamics Review* **22**(1): 51–71.

Weil HB, Etherton RL. 1990. System dynamics in dispute resolution. In *Proceedings of the 1990 International System Dynamics Conference*, Utrecht, The Netherlands.

Williams TM. 1999. Seeking optimum project duration extensions. *Journal of the Operational Research Society* **50**: 460–467.

Williams T. 2002. *Modelling Complex Projects*. Wiley: Chichester.

Published online in Wiley InterScience
(www.interscience.wiley.com) DOI: 10.1002/sdr