NPS-PM-06-084

EXCERPT FROM THE PROCEEDINGS

OF THE

THIRD ANNUAL ACQUISITION RESEARCH SYMPOSIUM

MANAGING TIPPING POINT DYNAMICS IN SINGLE DEVELOPMENT PROJECTS

Published: 30 April 2006

by

David N. Ford and Tim Taylor

3rd Annual Acquisition Research Symposium of the Naval Postgraduate School:

Acquisition Research: Creating Synergy for Informed Change

May 17-18, 2006

Prepared for: Naval Postgraduate School, Monterey, California 93943

ACQUISITION RESEARCH PROGRAM
GRADUATE SCHOOL OF BUSINESS & PUBLIC POLICY
NAVAL POSTGRADUATE SCHOOL

The research presented at the symposium was supported by the Acquisition Chair of the Graduate School of Business & Public Policy at the Naval Postgraduate School.

To request Defense Acquisition Research or to become a research sponsor, please contact:

NPS Acquisition Research Program
Attn: James B. Greene, RADM, USN, (Ret)
Acquisition Chair
Graduate School of Business and Public Policy
Naval Postgraduate School
555 Dyer Road, Room 332
Monterey, CA 93943-5103
Tel: (831) 656-2092
Fax: (831) 656-2253
E-mail: jbgreene@nps.edu

Copies of the Acquisition Sponsored Research Reports may be printed from our website www.acquisitionresearch.org

Conference Website:
www.researchsymposium.org

Proceedings of the Annual Acquisition Research Program

The following article is taken as an excerpt from the proceedings of the annual Acquisition Research Program. This annual event showcases the research projects funded through the Acquisition Research Program at the Graduate School of Business and Public Policy at the Naval Postgraduate School. Featuring keynote speakers, plenary panels, multiple panel sessions, a student research poster show and social events, the Annual Acquisition Research Symposium offers a candid environment where high-ranking Department of Defense (DoD) officials, industry officials, accomplished faculty and military students are encouraged to collaborate on finding applicable solutions to the challenges facing acquisition policies and processes within the DoD today. By jointly and publicly questioning the norms of industry and academia, the resulting research benefits from myriad perspectives and collaborations which can identify better solutions and practices in acquisition, contract, financial, logistics and program management.

For further information regarding the Acquisition Research Program, electronic copies of additional research, or to learn more about becoming a sponsor, please visit our program website at:

www.acquistionresearch.org

For further information on or to register for the next Acquisition Research Symposium during the third week of May, please visit our conference website at:

www.researchsymposium.org

THIS PAGE INTENTIONALLY LEFT BLANK

Managing Tipping Point Dynamics in Single Development Projects

Presenter: David N. Ford, PhD, PE, is an Assistant Professor in the Construction Engineering and Management Program in the Zachry Department of Civil Engineering, Texas A&M University. In addition to teaching, he researches development project strategy, process and policy design, and flexibility in projects. Prior this work, Dr. Ford was on faculty in the Department of Information Science at the University of Bergen, Norway, where he researched and taught in the System Dynamics Program. Dr. Ford earned his PhD degree from the Massachusetts Institute of Technology and his Master's and Bachelor degrees in Civil Engineering from Tulane University. He has over 14 years of industry and government experience in engineering and project management.

Presenter: Tim Taylor is a PhD Candidate in Construction Engineering and Management in the Zachry Department of Civil Engineering at Texas A&M University. He holds a BS and MS in Civil Engineering from the University of Kentucky. His research interests include strategic project management, real options, loop dominance analysis, and project robustness.

David N. Ford
Zachry Department of Civil Engineering,
Texas A&M University,
College Station, Tx 77843-3136.
Phone: (979) 845-3759.
Fax: 979-845-6554.
Email: davidford@tamu.edu

Abstract

Previous system dynamics work models the tipping of a series of product development projects into fire-fighting mode in which rework overwhelms progress. Similar dynamics also threaten the performance of individual development projects. The current work extends previous tipping point dynamics research to single projects and demonstrates how a simple, common feed-back structure can cause complex tipping point dynamics, trap projects in deteriorating modes of behavior, and cause projects to fail. Basic tipping point dynamics in single projects are described, analyzed, and demonstrated with the model. Previous researchers have recommended dynamic resource allocation policies to improve project performance threatened by tipping point dynamics. Several strategies for managing projects near tipping points were tested. Policies that were successful in preventing tipping point-based project failure include forecasting demand-based resource policies, policies that provided flexible resource adjustments, and policies that adjusted project deadlines based upon project performance.

Keywords: resource allocation, nuclear plant construction, project management, tipping point, robustness, system dynamics

Introduction

Although development projects are pursued to add value for their developers or users, many projects fail (Evans, 2005; Matta & Ashkenas, 2003; Wells, 1999). Project failure can take many forms, including schedule and cost overruns and unacceptable quality. Project failure is relatively easy to identify if the final product grossly fails to meet performance targets (e.g., some of NASA's Mars probes) or if development stops before a product is completed (e.g., the US Department of Energy's Supercollider project). But some projects that are completed should also be considered failures. An example is the Channel Tunnel (the "Chunnel") that connects England and France. While the Chunnel is arguably one of the great engineering achievements of the last century, its final cost of $17.5 billion was more than double the original estimate of $7.2 billion (Kharbanda & Pinto, 1996). Chunnel usage is below the level estimated in the project's feasibility study, and even the most optimistic estimates predict that the Chunnel will not be profitable in the next 10 to 20 years (Kharbanda & Pinto, 1996). Although a technical marvel, the Chunnel failed to meet two of its fundamental goals: finish with budgeted funds and produce a financially viable product. Failure of these large projects can have dire consequences for all parties associated with the project.

Project management research has identified many factors that can lead to project failure including overestimation of benefits (Evans, 2005), poor stakeholder analysis (Paul, 2005), and errors (Busby & Hughes, 2004). Despite considerable research into these factors, clearly identifying project failure is difficult. Comparing differences between project performance and targets is a standard means of measuring project success or failure. But variations of final project performance from targets can be poor measures if targets are flexible. For example, US Department of Energy projects are not allowed to exceed Congressionally approved budget targets. So, targets are revised based on final performance, even in cases of gross cost overruns. If performance relative to original targets is a measure of project success or failure, some Department of Energy projects that meet final targets should be considered failures (USGAO, 1996, 1997). Some organizations explicitly label such projects as failures. For example, as part of development improvement efforts, one organization known to the authors labeled a set of completed projects that exceeded their cost or schedule targets by 20% or more as "wrecks" (as in "train wrecks"). A clear, inclusive definition of project failure is needed to study the performance of projects. Changes over time in the work remaining to be completed can provide an improved metric. Although these project backlogs are intended to generally decrease over time, they can stagnate or grow. Projects with backlogs that increase continuously over significant periods of time ultimately lead to failures to meet original project targets and may be terminated. The current work defines a project as a failure when its backlog grows continuously over an extended period of time.[1]

The continuous growth of project backlogs over time can be attributed to many different dynamic factors. Dynamic causes identified through system dynamics include a lack of knowledge transfer between projects (Cooper et al., 2002), rework (Cooper, 1993a,b,c) and concealing rework (Ford & Sterman, 2003b), schedule pressure (Cooper, 1994; Ford & Sterman, 2003a), and "fire-fighting" (Repenning, 2001). A complete dynamic hypothesis of development project failure would include unrealistic performance targets and how negative feedback loops that describe responses to schedule, budget, and other pressures can trigger fatal reinforcing loops through productivity losses, overstaffing, inadequate training, and other project behaviors. Other exogenous changes that slow progress, degrade performance, and can lead to failure (e.g., increased regulation, scope changes, temporary work stoppage) would

[1] Active projects that stagnate, with no change in project backlog over time, are also considered failures but are less common. As will be shown, these conditions can be unstable, and stagnant projects are likely to shift behavior modes into an increasing or decreasing project backlog.

provide the bases for additional hypotheses. The dynamic structure would also include the amplification of impacts due to delays in discovering rework that allow problems to be passed among development phases. These and other causes of project failure have been used in system dynamics practice, and several have been addressed in the literature.

The current work focuses on how a particular dynamic structure, tipping points, can cause a common project feature, ripple effects, to generate rework and project failure. A development project's ripple effects are the secondary or tertiary impacts of a change. Thomas and Napolitan (1994) identify indirect changes due to ripple effects due to work interdependency in construction projects as an important cause of project failure. They estimate the impacts on labor efficiency in some projects to be seven times larger than the impacts of direct changes. Ripple effects can be triggered by many unplanned events or conditions, including the exogenous factors described above. Likewise, ripple effects can have multiple types of impacts, including creating more work, requiring rework in previously correct work, and reducing productivity. We focus on the work effort created by ripple effects and disaggregate that effort into two forms, contamination and adding new tasks[2]. Contamination is work required in part of the *existing* project scope that is created due to rework being discovered in a different portion of the project. For example, if after a reinforced concrete column was poured, the inspectors discovered that the reinforcing steel used was too small, part of the beams above and below the column might have to be demolished in order to replace the column. Replacing the column (rework) requires reworking the beams even though the beams were not otherwise defective. The column rework contaminated the adjacent beams, but did not add any *new* activities to the project. In contrast to contamination, adding new tasks, as used here, creates development activities *beyond* the project scope due to rework required on portions of the existing project scope. In the column example, temporary shoring required to support the upper beams while the column is replaced would be new tasks. Rework on previously created new tasks can also contaminate and add more new tasks. For example, inadequate temporary shoring of the beams in the column example could damage adjacent floors (contamination) and require more shoring for floor repairs (more new tasks). The critical difference between contamination and adding new tasks is that contamination creates more rework within the existing project scope or previously added tasks, while adding new tasks creates development activities that were not previously a part of the project. The current research focuses on adding new tasks because it can be difficult to identify during the course of a project when created by rework and, as will be shown, can cause challenging project behavior and failure.

Tipping points are one explanation of bifurcated system behavior such as project backlogs that diminish and lead to success or grow and lead to failure. A tipping point is a threshold condition that, when crossed, shifts the dominance of the feedback loops that control a process (Sterman, 2000). Systems tend to remain stable as long as conditions remain "below" the tipping point, and controlling feedback is dominant (Sterman, 2000, p. 306). But when conditions cross the tipping point, behavior can become (temporarily) unstable and, in the case of projects, lead to failure. Social physiologists have used tipping points to describe an unexpected spread of disease, a dramatic change in the crime rate in a city, and an increase in the number of teenage smokers despite a campaign of increased awareness (Gladwell, 2000).

[2] As used here, *scope* refers to the tasks, measured in work packages, that, when approved and released, provide a specified performance; *work* is an amount of development effort, also measured in work packages. Rework and adding new tasks cause the work required to complete the project to exceed the scope.

System dynamics can be used to elucidate tipping points and their impacts on systems in several ways: 1) by specifying, formalizing, and explaining structures that create tipping points, 2) by describing behaviors resulting from tipping points, and 3) by developing policies for managing systems with tipping points. Here we investigate whether a combination of a tipping point structure and ripple effects can explain the failure of some large, complex development projects.

The current work examines the generation of tipping point dynamics due to ripple effects in single project development systems and tests strategies and policies for resistance to project failure. Challenges posed by tipping points in single projects are discussed next. Two examples of tipping point failure in the nuclear power industry are presented. Then, a model of a single-product development project is used to examine the impact of ripple effects on project success. Exogenous, endogenous, and combined drivers of behavior modes are followed by testing policies for tipping point solutions, and the use of robustness as a measure of potential project failure is tested. The conclusion discusses managerial implications and research opportunities.

Project Management Challenges near Tipping Points

Complex development projects are difficult to manage because of the dynamic nature of project systems (Lyneis et al., 2001). A project manager's ability to understand these non-linear feedbacks is limited. Most project management tools available, such as the critical path method, are linear and cannot adequately predict the effect increased rework and added new tasks has on a project. Systems dynamics is more suited to the modeling of development dynamics. Such models must include iterative flows of work, distinct development activities and available work constraints both within and among development phases. The existing system dynamics models of projects (which include process structures) have focused on the roles of two development activities. Cooper (1994, 1993a,b,c, 1980) first—and several researchers subsequently (e.g. Kim, 1988; Abdel-Hamid, 1984; Richardson & Pugh, 1981)—modeled two development activities by distinguishing between initial completion and rework. This distinction allows the effect of rework on a project to be studied.

Similar structures and conditions may drive some individual development projects. Many product development projects are managed largely in isolation from other projects and can fail due to dynamics solely within or near a single project. Therefore, the explanation of tipping point impacts on project performance needs to be expanded to single project design and management. The current work extends the multi-project work by Repenning (2001) and Black and Repenning (2001) to single development projects. We use a system dynamics project model to examine the effects of ripple effect-induced tipping point dynamics on single project behavior and performance. This work contributes a new explanation for the failure of some large, individual development projects. The understanding of development project dynamics is advanced by proposing and initially testing the ability of a specific project structure to generate tipping point dynamics. That understanding is the basis for proposing and testing policies for preventing or managing projects that are vulnerable to failure due to tipping point dynamics.

Tipping Point Dynamics in Nuclear Power Plant Construction

The first commercial nuclear plant to come on-line in the United States was Dresden 1, located in Illinois, in 1959 (NRC, 1982). Between 1959 – 1969, twelve nuclear plants were completed with an average construction duration of 46 months (NRC 1982). Between 1970 –

1981, the average duration nearly tripled, reaching 131 months in 1981 (NRC, 1982). While the plants constructed during this time were higher capacity units (i.e., bigger) than earlier projects, most researchers identify the ever increasing (and ever changing) number of governmental regulations imposed on nuclear plants as the root cause for cost and duration increases in nuclear plant construction (Lake, 2002; Kharbanda & Pinto, 1996; Feldman et al., 1988; Lillington, 2004; Friedrich et al., 1987). Examining construction records from the Nuclear Regulatory Commission from this period provide two examples of possible tipping point dynamic failure.

The first example of a project that crossed this tipping point is the Tennessee Valley Authority's (TVA) Watts Bar nuclear power plant units 1 and 2. TVA began the construction of the Watts Bar facility in December of 1972 (NRC, 1982). Originally the facility was to consist of two 1165 MW units that were to both be on-line by the middle of 1977 (NRC, 1982). However, as Figure 1 shows, the two units were unable to meet the planned deadline. By mid 1977, Unit 1 was 57% complete and Unit 2 was 49% complete. In May of 1974, the TVA reported delays due to the redesign of the reactor containment vessel to accommodate higher pressures, an inability to obtain redesigned anchor bolts and reinforcing rods, and increased time to erect steel plates that were thicker than the original specifications (NRC, 1982). The work created by the problems beyond the original scope (e.g., additional anchor bolts or steel plates) are evidence of adding new tasks. Work was halted in 1980 for five years to address worker safety concerns with the design of the plant (Lee, 1995). To address these concerns, the TVA spent nearly one million man hours reviewing the design of the plant (Lee, 1995). This review lead to the replacement of nearly three million feet of cable, 8,000 pipe supports, and 25,000 conduit supports (Lee, 1995). The TVA canceled Unit 2 in 1995 with the unit 61% complete (Nuclear Engineering International, 1995). The TVA estimated that it would cost more than the $1.7 billion already invested in Unit 2 to complete the unit. When Unit 1 finally came on line in 1996, the TVA had invested nearly $7 billion dollars in the facility (Lillington, 2004). The decrease or stagnation in the fraction of the total project scope that has been completed (right side of Figure 1) is a characteristic behavior of projects experiencing strong ripple effects.

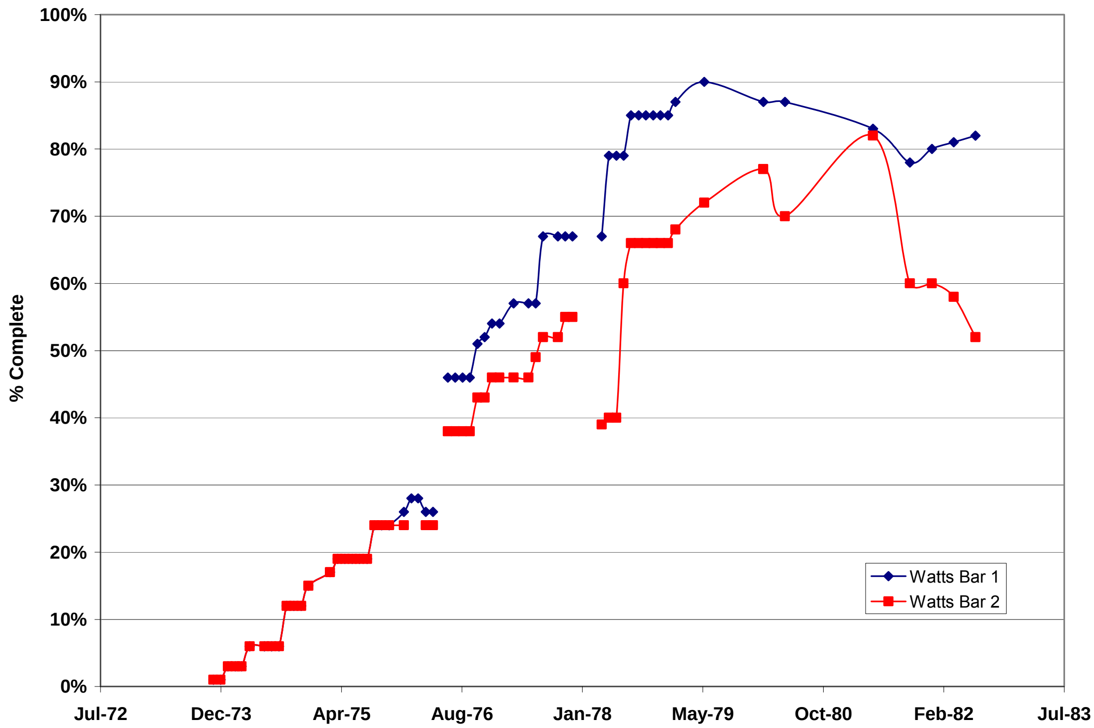

Figure 1. Watts Bar Construction Progression (1973-1982) (NRC, 1982)

The second example of tipping point failure is Philadelphia Electric's Limerick nuclear power plant. Construction of the two 1065 megawatt units began in June of 1974. The construction schedule at the issuance of the construction permit called for Unit 1 to be competed in April of 1979 and for Unit 2 to be completed in September of 1980 (NRC, 1982). The total estimated cost for both units in 1974 was $1.2 billion (Days & Sellers, 1985). As Figure 2 shows, both units were well behind schedule at their respective planned completion dates (Unit 1 at 48% complete and Unit 2 at 36% complete) (NRC, 1982). Unit 1 finally came on-line in August of 1985, five and a half years behind schedule with a final cost of $3.8 billion (Days & Sellers, 1985). Construction of Unit 2 was halted in July 1982 by order of the Pennsylvania Public Utility Commission due to escalating costs (NRC, 1982).

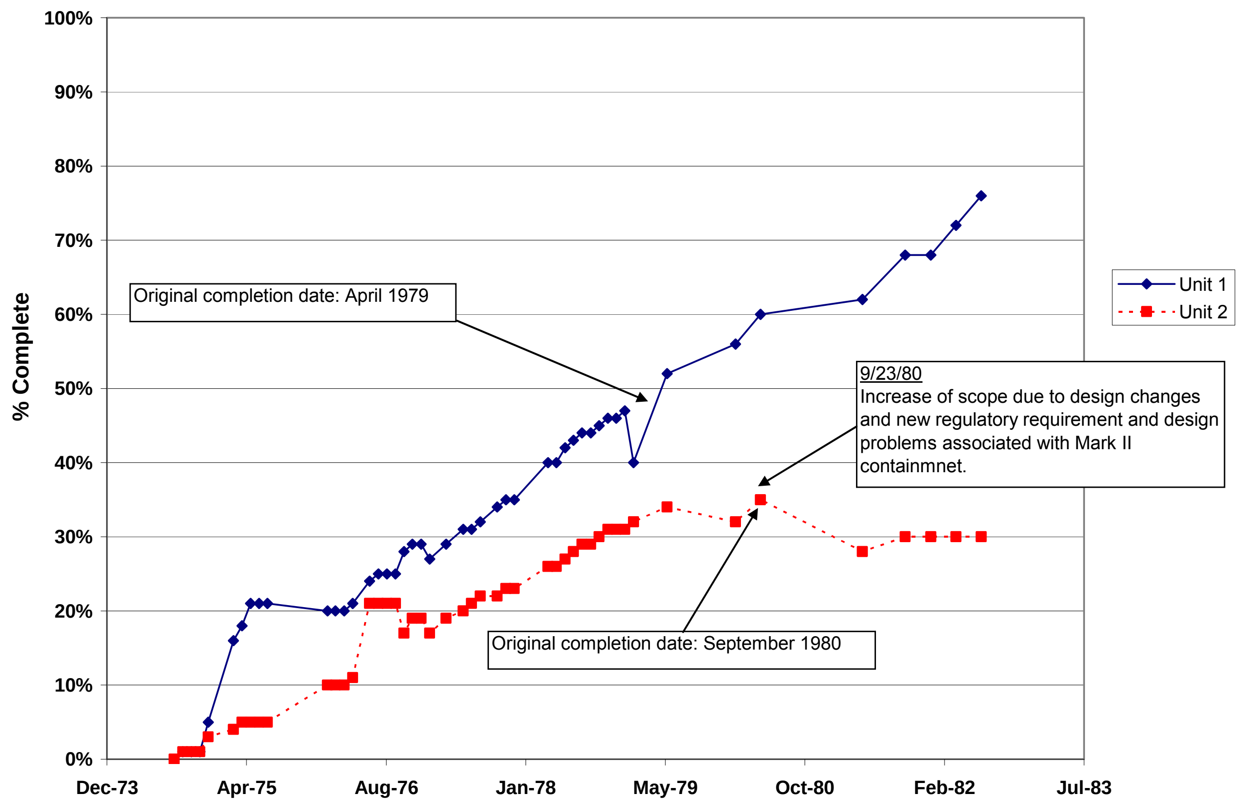

Figure 2. Limerick Nuclear Power Plant Construction (1974-1982) (NRC, 1982)

The Philadelphia Electric company attributed at least part of the cost and schedule problems to added new tasks and rework, two factors which Taylor and Ford (2006) showed capable of generating tipping point dynamics. In a September 1980 report submitted to the NRC, the estimated completion date was increased by two years for Unit 2 due to an "increase of scope [added new tasks] due to design changes and new regulatory requirements [rework]" (NRC, 1982). The degrading backlog behavior pattern is displayed on Unit 2 in Figure 2 between May 1979 and October 1980 as the percent complete begins to decrease. The Limerick plant was not the only plant to experience problems. A survey of senior managers at a firm specializing in nuclear plant construction revealed that nearly all surveys credited regulatory changes as the major cause for delays in both design and construction of nuclear power plants (Arditi & Kirsinikas, 1985).

The failure of Watts Bar and Limerick are not isolated incidents of nuclear plant project failure. An investigation of 45 nuclear power plants under construction between 1973 and 1982 revealed that only 2 of the plants finished at or before their original deadline. Figure 3 shows the frequency of schedule overrun for the 45 nuclear plants with the most frequent level of overrun between 100%-150% (NRC, 1982).

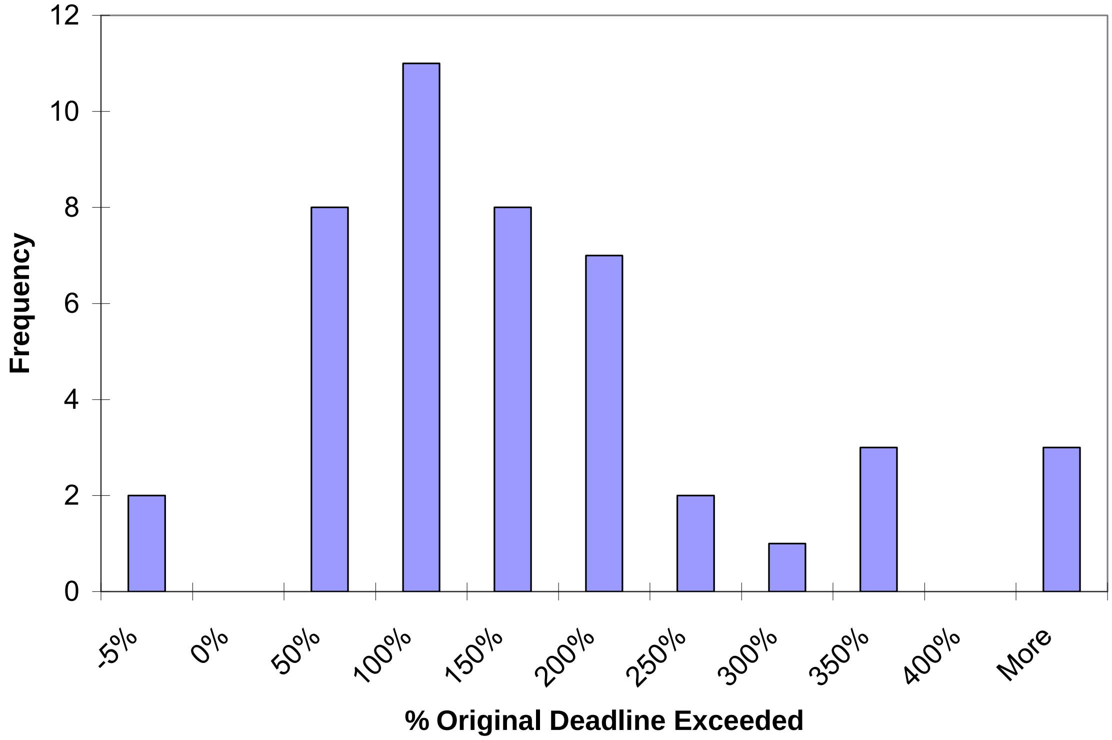

Figure 3. Schedule Overrun for Nuclear Power Plants (1973-1982) (NRC, 1982)

Further analysis of 52 nuclear plants under construction during 1977-1982 reveals that for 1370 total progress reports, 513 (37%) showed no net progress (i.e., stagnant backlogs) between report periods and 71 (5%) showed declining progress (e.g., increasing backlogs)—again behavior that could indicate the presence of tipping point dynamics.

A specific example of the effect of changing governmental regulation in nuclear plant construction is the changing requirements for pipe supports. In 1971, a new regulation was adopted that required all pipes within a nuclear power plant to be supported (Aron, 1997). This included pipes within the reactor containment building (the large concrete dome seen at all US nuclear plants). Designers failed to take into account the effect this change would have on plant design. One such example is the Shoreham plant in Long Island, New York.

Construction began in 1972 and was to be completed in the first quarter of 1977. In September of 1977, the expected duration was increased by over 12 months due to material shortages, labor productivity, and "design changes due to regulatory requirements" (NRC, 1982). The original estimate of $217 million was well short of the actual $5 billion cost when the plant was completed (Aron, 1997). Construction activities were completed in the mid 1980's, but the plant never came on-line due to political pressure over concerns of evacuating Long Island in the event of an accident (Aron, 1997).

Although the plant did not begin construction until 1972, the design was already completed and approved before the new pipe support regulations took effect. According to a former Vice-President of the architect/engineering/construction firm building the Shoreham plant, during construction, the pipe supports had to be designed and installed (Reinschmidt, 2005). As these supports were outside the initial scope, they provide an example of adding new tasks.

Pipe support changes were not limited to the Shoreham plant. Friedrich et al. (1987) referred to the "reengineering and redesign" of pipe supports as a "frequently encountered event" in nuclear plant construction. Changing regulations along with changes in market conditions helped the economic viability of nuclear plants become suspect. A 1988 study (Feldman et al.) suggested that for most nuclear power plants under construction in the United States at the time, it would be more economical to either cancel the plants under construction, regardless of progress, or modify the plants to burn conventional fuel (coal, gas, or oil). This illustrates the potential large impact tipping point dynamics can have on single development projects.

A Simulation Model of Project Tipping Point Dynamics

Most traditional project-management models, such as the critical path method, are linear and cannot adequately predict the effects that increased rework, contamination, and the addition of new tasks have on projects. In contrast, systems dynamics is well suited to modeling development dynamics. System dynamics has a strong and established history of modeling development projects and has been successfully applied to a variety of project management issues, including failures in fast track implementation (Ford & Sterman, 1998), poor schedule performance (Abdel-Hamid, 1984), and the impacts on project performance of changes (Rodrigues & Williams, 1997; Cooper, 1980, 1993a,b,c) and concealing rework requirements (Ford & Sterman, 2003a).

The model is purposefully simple relative to actual practice to expose the relationships between tipping point structures, project behavior modes, and management. Therefore, although many development processes and the features of project participants and resources interact to determine project performance, only those features that describe a particular tipping point structure, project management policies, and the fundamental processes they impact are included. Simulated performances using different policies are, therefore, considered relative and useful for improving understanding and developing insights, but not sufficient for final policy design. Complete model equations and documentation are available from the authors or at http://ceprofs.tamu.edu/dford/.

The model consists of three sectors: a workflow sector (Figure 4), a resource allocation sector, and a schedule sector. The workflow sector is based on Ford and Sterman's (1998) structure of a development value chain with a rework cycle. Work is initially completed and moves from the initial completion backlog[3] (IC backlog) to the backlog of work requiring quality assurance (QA backlog). A fraction of the work checked by quality assurance is discovered to require change and moves into the rework backlog. Completed rework is returned to the QA backlog for checking again because rework can reveal previously hidden or create new change requirements[4]. The complement of the checked work found to require rework passes quality assurance, is approved, released, and adds to the stock of work approved and released (Work

---

[3] Development activity flows represent the completion of a development task. Therefore backlogs, as used here, include work in progress.

[4] This creation of additional rework is not contamination because it represents additional rework required in the same piece of work, not additional rework required in a different piece of work.

Released). Flows between the stocks of IC backlog, QA backlog, RW backlog, and Work Released can be constrained by either process rates or resources. Process rates assume infinite resources and are the amounts of work available divided by the minimum times required to perform a work package. Resource rates are the products of the quantities of resources assigned to each activity and resource productivity. See Ford and Sterman (1998) for a more detailed work flow model description and model equations.

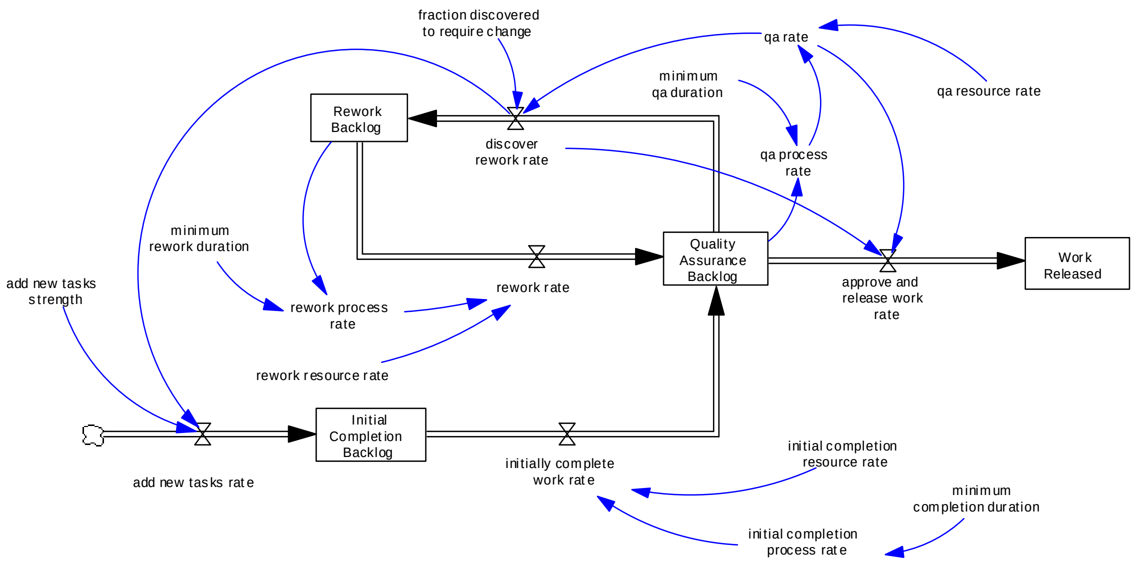

Figure 4. Work Flows for a Single Project System (based on Ford & Sterman, 1998)

A unique expansion of this model in the current work is the explicit modeling of adding new tasks in a tipping point structure. Adding new tasks creates work that is added to the IC backlog during a project. We assume that the amount of work created is proportional to the work discovered to require rework:

$$R_{nt} = (D_{rw}) (s_{nt}) \qquad (1)$$

Where:

R_{nt} - rate of adding new tasks due to ripple effects {work packages / week}

D_{rw} - discover rework rate {work packages / week}

s_{nt} – add-new-tasks strength {work packages created / work packages discovered, or dimensionless}

The add-new-task strength is a project characteristic that describes the amount of impact that reworked portions of the project have on the total work required to complete the project and, thereby, can be used to describe different project types. It is related to the amount of interdependence between project subsystems. For example, the strength between the foundation and superstructure components of a building would be high compared to the strength between the foundation and the heating system.

Resources are allocated among the initial completion, quality assurance, and rework activities proportionally based on the current demand for each of these activities. The desired fraction of resources for each activity is the size of the backlog compared to the project backlog (ICbacklog+QAbacklog+RWbacklog). For example, if resource productivities are equal and the current RW backlog is 40% of the current project backlog, the desired portion of the available resources to be allocated to the rework activity is 40%. Applied resource fractions are delayed with a first order exponential adjustment toward the desired fractions to reflect reallocation delays.

Schedule pressure is common in development projects. Increased rework is a side effect of schedule pressure that can degrade project performance (Cooper, 1994; Graham, 2000; Ford & Sterman, 2003b).[5] As a project approaches a fixed deadline, schedule pressure increases; developers increase the pace of work to meet the deadline. This increases the risk of work being completed incorrectly. In the schedule sector, pressure increases with the time required to complete the project backlog (t_r) and decreases with the time available to complete the project backlog (t_a). To explicitly model the impacts of schedule pressure on tipping point dynamics, we disaggregate the rework fraction (f_{rw}) into the sum of a reference rework fraction (f_{rw-r}) and the schedule-induced rework fraction (f_{rw-s}). The reference rework fraction reflects basic project complexity. The schedule-induced rework fraction is the additional fraction of work requiring change due to schedule pressure. The schedule-induced rework fraction reflects mistakes made by developers due to pressures to meet the project deadline. This portion of the rework fraction is modeled as the product of schedule pressure and the sensitivity of the rework fraction to schedule pressure (s_{rw-s}). Forgoing the functions to limit values to 0-100%, the rework fraction becomes:

$$f_{rw} = f_{rw-r} + f_{rw-s} = f_{rw-r} + [((t_r / t_a)-1)\ (s_{rw-s})\ ] \qquad (2)$$

Where:

f_{rw} - rework fraction (dimensionless)

f_{rw-r} - reference rework fraction {dimensionless}

f_{rw-s} - rework fraction due to schedule pressure {dimensionless}

t_r - time required to complete project backlog {weeks}

t_a - time available to complete project backlog {weeks}

s_{rw-s} – sensitivity of rework to schedule pressure {dimensionless}

---

[5] Schedule pressure can have multiple beneficial and detrimental impacts on project performance which can be modeled with additional feedback loops (see Ford, 1995 for examples). The current work models only the net effects of schedule pressure on rework and assumes the net effect is negative.

Figure 5 shows the work flow structure (Figure 4) and tipping point feedback structure. Feedback loop B1 (Project Progress) withdraws work from the rework cycle. The QA backlog increases due to initial completion and rework, causing the QA rate to increase as resources are shifted to quality assurance. Increasing QA increases the rate at which work is approved and decreases the QA backlog. This balancing loop drives the project to completion as the backlogs decline to zero. If no new tasks are added, B1 completes a project as quickly as processes and resources allow.

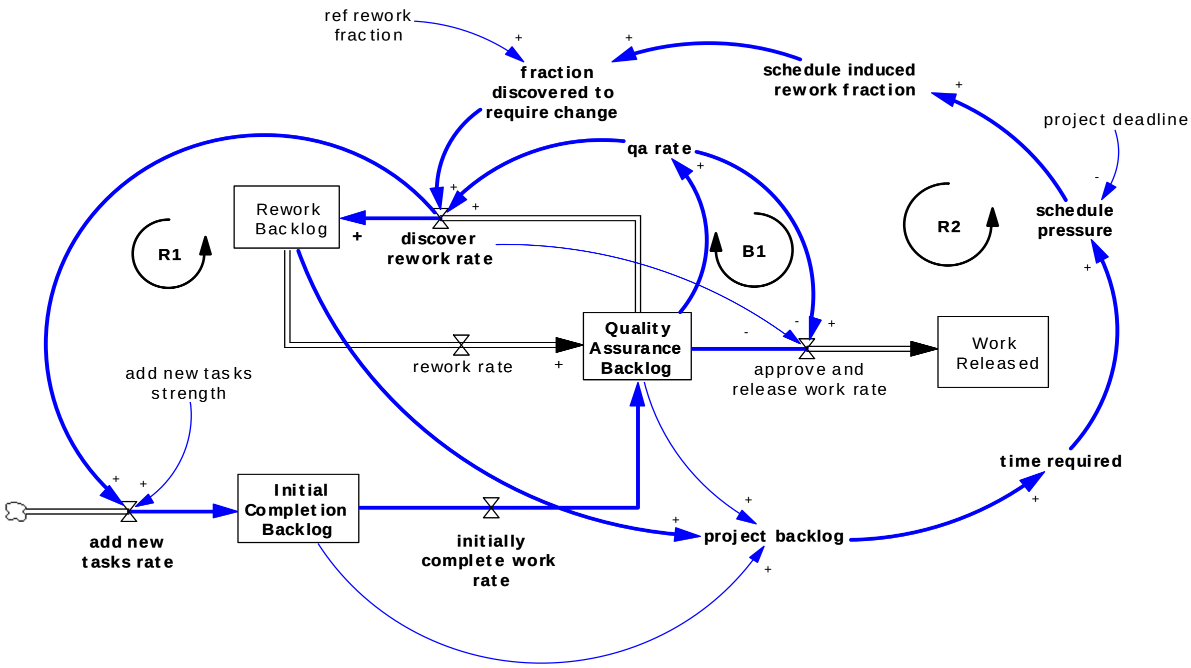

Figure 5. A Tipping Point Structure of a Single Development Project

Loop R1 (Add New Tasks) adds to the total work required to complete the project through increases in the discovery of rework and adding new tasks—increasing initial completion and, thereby, the QA backlog increases the QA rate, increasing the rate at which work is discovered to require rework. This increases the rate at which new tasks are added, thereby adding more work to the IC backlog. In the absence of loop B1 (e.g., if the rework faction = 100%) loop R1 increases the rework and project backlog infinitely, thereby degrading project performance to eventual failure.[6] Feedback loops B1 and R1 form a traditional tipping point structure that can dramatically change system behavior from being "under control" to being "out of control" due to a shift in feedback loop dominance from the balancing loop to the reinforcing loop. We show how, through exogenous manipulation of loop dominance, managers can regain control of projects with tipping point structures. Loop R2 (Schedule Pressure) can increase the strength of the Add New Tasks loop (R1) by increasing the rework fraction as

[6] The loop dominance analysis discussed here is consistent with the results of a more rigorous analysis performed using behavioral analysis presented in Ford (1999). See Taylor et al. (2005) for details.

described above. The resulting increase in a project's backlog increases the time required to complete the project, increasing schedule pressure. This increases the schedule-induced rework fraction and, thereby, the fraction discovered to require rework[7].

Model Testing and Typical Behavior

The model was tested using standard methods for system dynamics models (Sterman, 2000). Basing the model on previously tested project models and the literature improves the model's structural similarity to development processes and practices, as do unit consistency tests. Extreme condition tests were performed by setting model inputs, such as initial scope or total project staff, to extreme values and simulating project behavior. Model behavior remained reasonable. The model's behavior for typical conditions is consistent with previous project models and practice (e.g., the common "S"-shaped increase in work released over time shown in Figure 6). As a successful project progresses, the backlog initially decreases slowly as the value chain and rework structures fill with work, increases progress during stable production, and decreases to zero slowly as backlogs empty, indicating that the project is complete. Model behavior was also compared to actual project behavior as described by Ford and Sterman (1998; 2003b) and Lyneis et al. (2001) and found to closely match the behavior modes of actual projects.

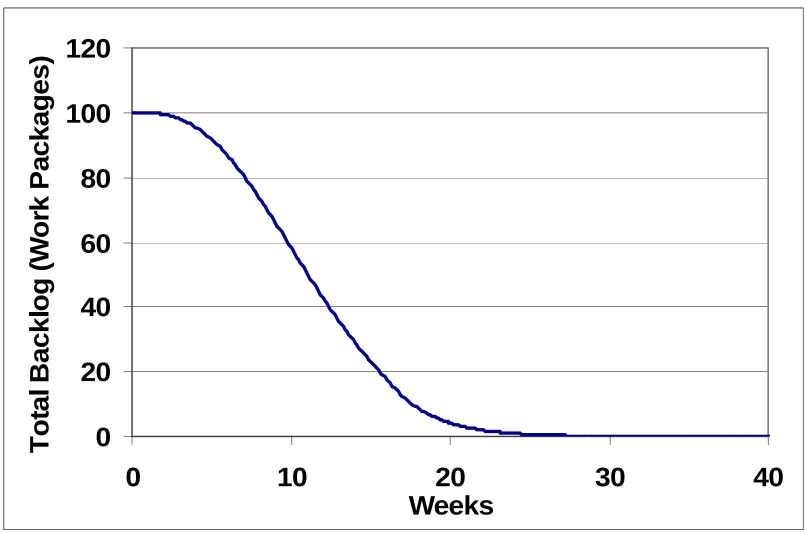

Figure 6. Typical Project Model Behavior

Limited project data prevented calibration to a specific project that experienced tipping point dynamics. Therefore, to test the ability of the model to replicate tipping point behavior modes, the model was calibrated with reasonable values to reflect a hypothetical project in

---

[7] Third and fourth reinforcing loops exist in which the IC backlog and IC rate increase the QA Backlog and, thereby, the QA rate and Rework Backlog. These backlogs also increase the project backlog. These loops perform like loop R2, but instead of increasing the project backlog through the IC backlog, they increase it through the QA and Rework Backlogs.

which the Add New Tasks and Schedule Pressure loops are active. The simulated behavior was compared to the behavior of the Tennessee Valley Authority (TVA) Watts Bar unit 2 project. The similarity between the actual project and simulated behavior modes in Figure 7 supports the model's ability to reflect a failure mode in nuclear power plant construction that could be caused by tipping point dynamics. Based on these tests, the model was assessed to be useful for investigating tipping point dynamics in single development projects.

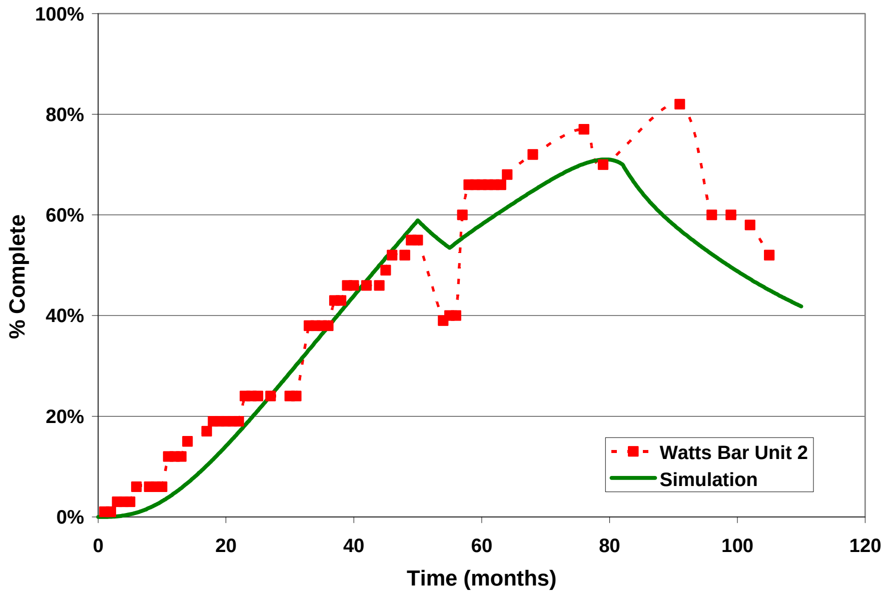

Figure 7. Tipping Point Behavior Mode Test

Similar to Repenning (2001), project progress is described with the project backlog as a fraction of the project's initial scope. Figure 8 shows the evolution of two types of projects. The horizontal axis shows project backlog in the previous time period; the vertical axis shows project backlog in the current time period. As an example of reading project behavior from the graph, the horizontal and vertical dashed lines show that in one project, the backlog was 80% of the scope in the previous time period and 72% in the current time period. All projects begin in the center of Figure 8, with backlog equals to their initial scope. *Improving* projects have decreasing backlogs and are reflected by conditions below the diagonal dashed line, when preceding project backlogs exceed the current project backlog. The behavior mode of the work released of the improving project in Figure 8 is the traditional "S-curve" common in project management literature. In contrast, *degrading* projects are reflected by conditions above the diagonal dashed line (when current project backlogs exceed previous project backlogs) and can theoretically have an ever-increasing backlog. The behavior mode of the project backlog of the degrading project in Figure 8 is an ever-increasing backlog. Successful projects end near the origin,[8] when there is no more project backlog. Failed projects approach the upper right corner of the graph, reflecting continuously increasing backlogs. A project that remains on any point along the

[8] The project simulation can reach the origin, when (PB_{t-1} , PB_t) = (0,0), but actual projects stop when the backlog first reaches zero, when (PB_{t-1} , PB_t) = (x,0) and x>0. This is represented in Figure 6 by a point on the horizontal axis close to the origin.

diagonal line has a constant project backlog and is stagnant (in net progress terms). An upper limit describing project failure has been arbitrarily set at 2, when work remaining to be completed is twice the original scope. All simulations in the current work reaching this limit have continuously increasing total backlogs and are considered failures. However, this limit may need to be adjusted for some projects.

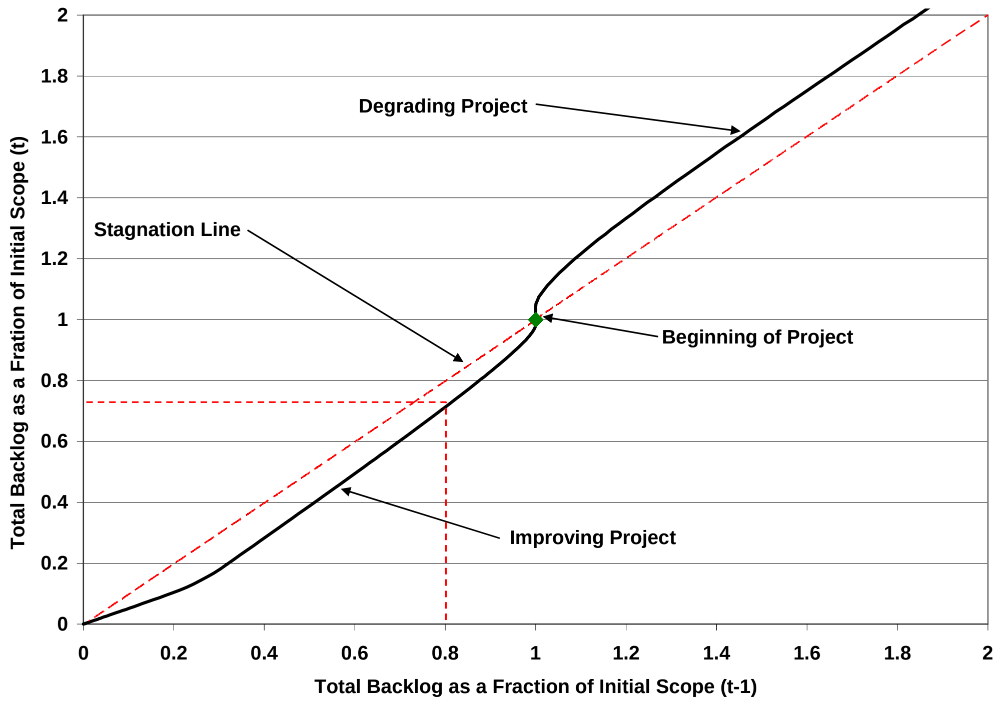

Figure 8. Evolution of Three Projects near a Project Tipping Point

In many projects, both loops B1 and R1 are active. In the improving project shown in Figure 8, the release loop (B1) is dominant; more work is being approved and removed from the rework cycle than is being added by new tasks through loop R1. For the degrading project, the Add New Task loop (R1) is dominant; more work is being added to the rework cycle than is being approved and released through loop B1. For stagnant projects (e.g. at the center of Figure 8), loops B1 and R1 are balanced; work is being removed from the rework cycle at a rate equal to the rate at which work is added to the rework cycle. The relationship between these two loops can be described using a tipping point.

Project Tipping Point Conditions

The tipping point is the condition between dominance by loop B1 (Figure 5) (leading to shrinking backlogs and project success) and dominance by loop R1 (leading to growing backlogs and failure). Adding new tasks adds work to the project backlog, and approving and releasing work withdraws work from the project backlog. Therefore, the tipping point occurs when the new task addition rate (R_{nt}) is equal to the rate at which work is approved and released. The rate at which work is approved and released is the complement of the QA rate that is discovered to require rework (D_{RW})[9]. Therefore, at the tipping point:

$$R_{nt} = R_{QA} - D_{RW} \quad (3)$$

Where:

R_{nt} - Rate of adding new tasks due to ripple effects {work packages/week}

R_{QA} - quality assurance rate {work packages/week}

D_{RW} - discover rework rate {work packages/week}

Temporarily using the aggregate rework fraction (f_{rw}), the rework discovery rate (D_{RW}) is the product the QA rate (R_{QA}) and the rework fraction. By substitution using equation (1), equation (3) becomes:

$$(s_{nt})(R_{QA})(f_{rw}) = R_{QA} - (R_{QA})(f_{rw}) \quad (4)$$

Simplification yields a description of the conditions that define the tipping point.

$$f_{rw}(s_{nt} + 1) = 1 \quad (5)$$

When the left-hand side of equation (5) exceeds 1, the project is degrading, when less than 1 the project is improving, and when equal to 1 the project is stagnant. A project can only remain at a tipping point (i.e. $f_{rw}(s_{nt}+ 1) = 1$) if loop B1 completes work at exactly the rate that loop R1 adds work to the project backlog. The project behavior will bifurcate to failure if loop R1 dominates or to success if loop B1 dominates. Therefore, the tipping point is an unstable equilibrium.

When the left-hand side of equation (4) exceeds 1 the project is degrading, when less than 1 the project is improving, and when equal to 1 the project is stagnant. The tipping point conditions are shown graphically in Figure 9.

---

[9] See Rahmandad (2005) for a similar project structure with constant addition of work and constrained work approval and release.

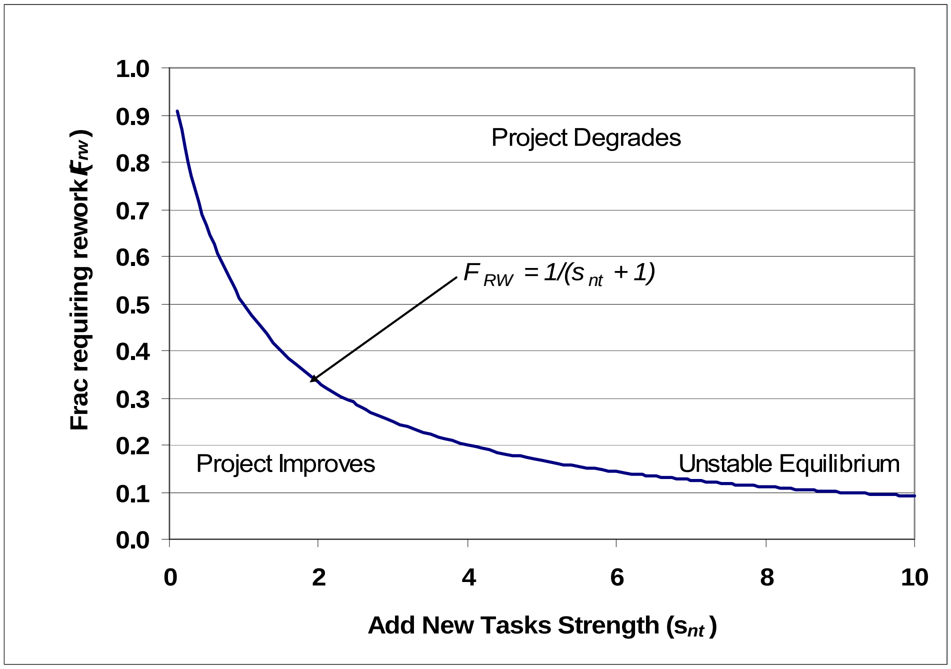

Figure 9. Basic Project Tipping Point Conditions

Figure 9 can be used to intuitively explain the behavior of projects near the tipping point. The total backlog of projects to the lower left of the solid line decreases, and the project improves. The total backlog of projects to the right of the solid line increases, and the project degrades. The solid line represents possible tipping point conditions. A project can only remain at a tipping point if the loop B1 completes work at exactly the rate as loop R1 adds work to the project backlog (Eq. 2). In the absence of forces to keep the project stagnate, small digressions from the tipping point conditions in either direction will cause the project to improve or degrade. If either loop dominates, total backlogs will increase (R1 dominates) to project failure or decrease (B1 dominates) to project completion. Therefore, the tipping point is an unstable equilibrium.

Project conditions that move across the tipping point conditions shown in Figure 9 experience a change in project behavior mode from increasing to decreasing or vice versa. The shape of Figure 9 reveals intuitive insights about project conditions that generate tipping point dynamics. The negative slope of the tipping point conditions line indicates that projects that have low add-new-task strength (s_{nt}) can tolerate a higher fraction of rework (F_{RW}) before degrading and projects with low rework fractions can tolerate higher add-new-tasks strengths. However, the tipping point relationship between add-new-tasks strength and rework fraction is not linear. A small increase in add-new-tasks strength greatly reduces the tolerable rework fraction. But as add-new-task strength increases, the tolerable rework fraction decreases more slowly, asymptotically toward a value of zero.

Project Trajectory Reversal and Schedule Pressure

We next investigate projects that begin on one side of the tipping point but, due to endogenous or exogenous influences, are pushed past the tipping point and reverse their behavior mode from improving to degrading or visa versa. As used here, project trajectory reversal is when the status of a project initially improves but later degrades and eventually fails (i.e., when an improving, “good,” project degrades, “goes bad”) or vice versa. Project trajectories that are monotonically improving or monotonically degrading (e.g., Figure 8) do not describe trajectory reversal. However, our study of large complex construction projects such as nuclear power plants indicate that trajectory reversal is an important issue. If project resources and productivity are limited and fixed, the basic project tipping point structure described above cannot endogenously simulate projects with trajectory reversal. This is because the structure lacks a mechanism to shift feedback loop dominance from loop B1 to R1. Exogenous influences, additional endogenous dynamic structures, or both, are required to propel projects beyond the tipping point and reverse their trajectory.[10]

Exogenous Influences on Tipping Point Dynamics

Exogenous factors can influence the rework fraction or add-new-tasks strength, such as changes in project scope during construction or, as with the case of the nuclear plant, changes in requirements. An inspection of equation (5) shows that, if a project starts far enough away from its tipping point (e.g. $F_{RW}\ (s_{nt}+1) <<1$) and the increases in the rework fraction and add-new-tasks strength are small enough, that the project does not cross the tipping point and behaves essentially like a monotonically improving project. However, if the magnitude of the changes is large enough, the project could be pushed past the tipping point, causing a project that initially improved to reverse its trajectory and degrade. However, as will be demonstrated next, pushing a project beyond its tipping point is not always sufficient to trap the project there and cause project failure.

Figure 6 shows the behavior of a project that begins with $F_{RW} = 0.2$ and $s_{nt} = 1$. Applying equation (5) ($F_{RW}\ (s_{nt}+1) =0.2(1+1) =0.4<1$) places the project on the improving side of the tipping point (pt. 1 in Figure 10). The project progresses towards completion until, at week 10 the rework fraction was exogenously raised to 0.6 to reflect a new but temporary problem that the development team must address. The tipping point conditions jump to 1.2, pushing the project quickly past the tipping point (pt. 2 in Figure 10). The project degrades, and the project backlog increases. The project continues to degrade until the rework fraction is exogenously returned to the original condition and the tipping point conditions return to their original level (pt. 3 in Figure 10). Once the project is operating below the tipping point again, it begins to reduce the project backlog (pt. 4 in Figure 10), thus improving to completion (pt. 5 in Figure 10). This demonstrates how improving projects subjected to large but temporary exogenous increases in the rework fraction, add-new-tasks strength, or both can be pushed beyond their tipping point and begin to degrade. But, barring structures that prevent a full and immediate recovery of those factors, when the exogenous change is removed, the project crosses the tipping point again and can improve again.

[10] Delays that can also cause shifts in feedback-loop dominance by temporarily constraining a strong loop are not addressed here.

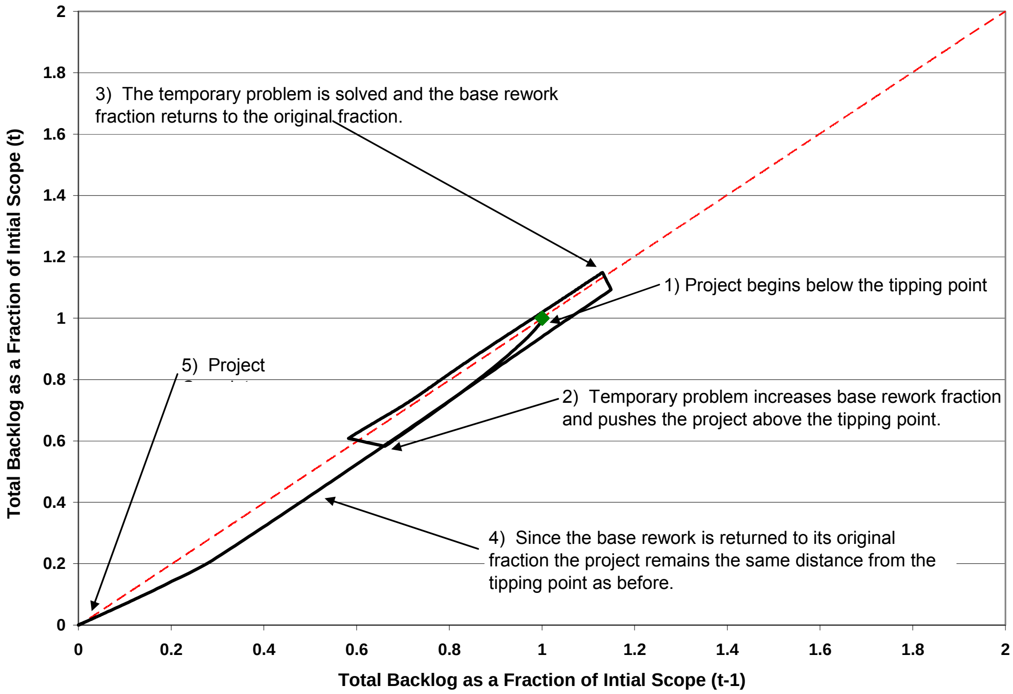

Figure 10. Project Exogenously Pushed beyond the Tipping Point

As modeled above, a sustained exogenous impact or another dynamic structure is required to cause an improving project to both reverse its trajectory and fail. In contrast to the behavior in the example above and in Figure 10, permanent exogenous changes in the rework fraction, add-new-tasks strength, or both that keep the project conditions beyond the tipping point (i.e. $F_{RW}(s_{nt}+1)>1$) generate project failure. Simulations not shown here for brevity verify these results.

A side-effect of the temporary problem in the example above is that the project will have a longer duration than it would have if the rework fraction had not increased. The time projects spend beyond the tipping point increases the total required work. If resource quantities and productivity is limited, this can cause projects to be completed far later than without the trajectory reversal. But, given enough time, the project will finish. This may be a partial explanation of projects that are very difficult to terminate and have very poor schedule performance (such as the Department of Energy projects described previously). In other cases, economic or other types of deadlines may cause these projects to be terminated, such as with nuclear plants that were never completed (Nuclear Engineering International, 1995).

Endogenously Influenced Tipping Point Project Failure

Some projects reverse their trajectory from improving to degrading and fail with continuously increasing project backlogs due to temporary problems. This suggests that temporary problems influence projects after the problem is resolved in ways that can cause failure. Setting aggressive deadlines is common in development projects. This generates schedule pressure, which can cause performance problems in development projects (Lyneis et al., 2001; Ford & Sterman, 2003a). Here, we investigate the impacts of schedule pressure due to aggressive deadlines on project performance through adding new tasks. Figure 11 shows the behavior of two projects (A and B) with different deadlines and, therefore, different amounts of schedule pressure. Without schedule pressure (feedback loop R2 inactive), the two projects finish in 25 weeks. The expected duration for project A has been reduced by 20% (20 weeks instead of 25 weeks). Project B has had its expected duration reduced by 28% (18 weeks). The interaction of schedule pressure and the tipping point have a dramatic impact on project performance. Project A remains on the improving side of the tipping point and finishes, but schedule pressure pushes Project B past the tipping point, causes trajectory reversal, and leads to failure. Simulations verify that the amount of schedule pressure that can be absorbed without trajectory reversal is related to the distance the project starts away from its tipping point conditions. These simulations demonstrate that projects can absorb safely some schedule pressure, but that in the presence of a tipping point structure, too much schedule pressure can cause projects to fail. The added new tasks–schedule pressure reinforcing loop provides an endogenous explanation for how projects that begin in conditions that can lead to success can become trapped beyond the tipping point, degrade, and fail.

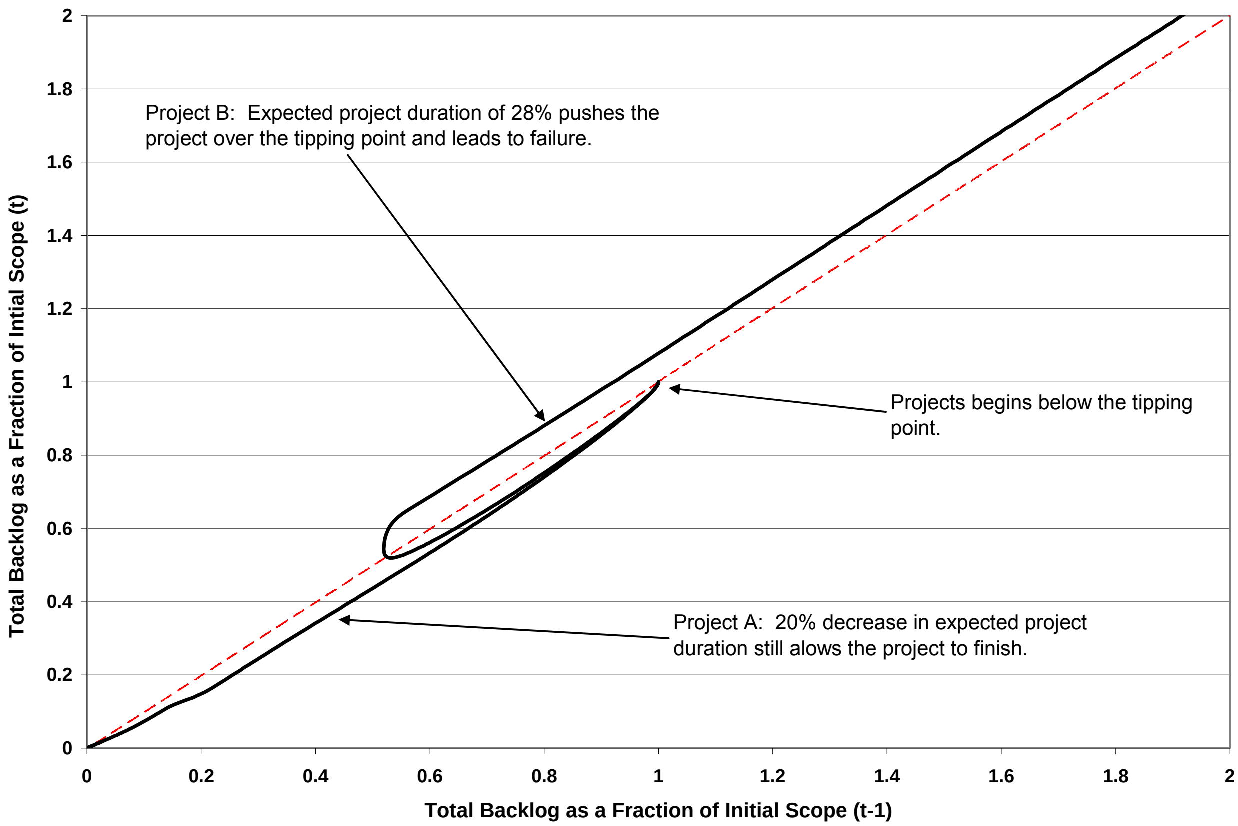

Figure 11. Effect of Schedule Pressure on Project Performance Mode

Compound Project Failure

Most development projects experience temporary problems, and many have aggressive deadlines. In these cases, as shown above, development projects can be doomed or likely to fail due to a tipping point structure. However, projects that can succeed despite temporary problems but do not are of particular interest to development project managers because they provide opportunities for improvement. Schedule pressure can trap projects that would finish (under normal circumstances) beyond the tipping point and drive them to failure. Figure 12 describes such a project. When applied individually, a temporary problem (Figure 10) or moderate schedule pressure (Project A in Figure 11) do not initiate permanent project degradation to failure. However, their combined impacted is enough to permanently push the project past the tipping point.

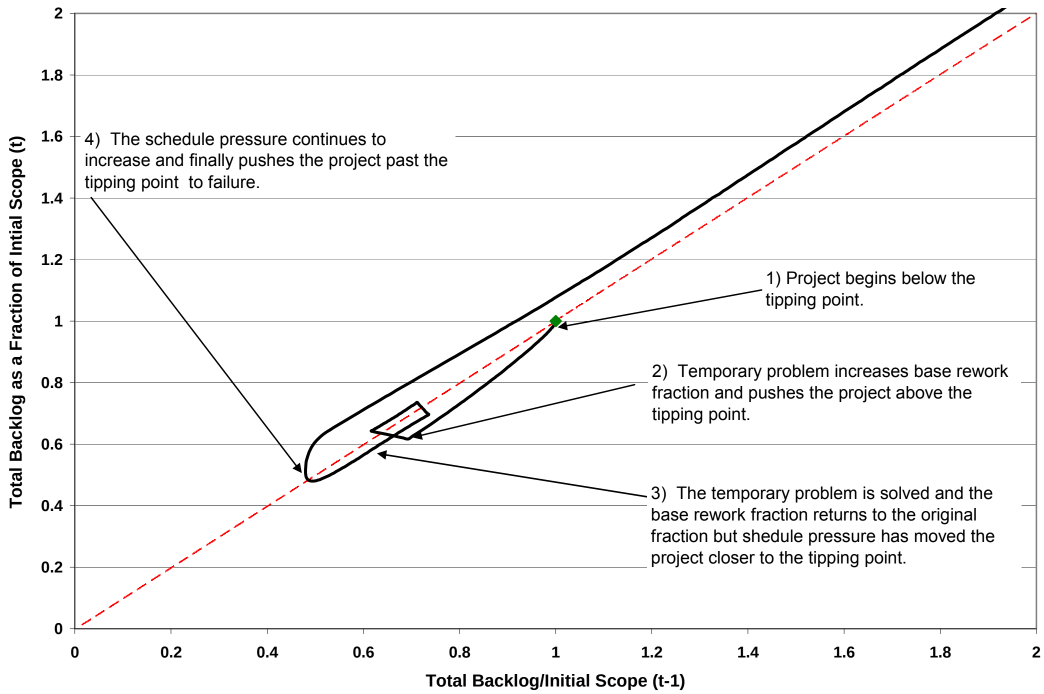

Figure 12. Interactive Impact of a Temporary Problem and Schedule Pressure on a Project

Consider a development project with an aggressive deadline that experiences an unexpected problem that temporarily increases the rework fraction. The project (see Figure 12) begins below the tipping point (pt. 1) and improves. In week 10, an exogenous temporary problem is encountered that pushes the project over the tipping point (pt. 2). The project begins to build project backlog, degrade, and increase schedule pressure. When the problem is resolved and the temporary increase in the rework fraction is removed, the project dips below the tipping point and begins to improve again (pt.3), but remains closer to the tipping point line than its previous position. This is due to the increased project backlog generated by the temporary problem; this increases the schedule pressure and, therefore, the rework fraction and added new tasks. In contrast to the project without schedule pressure, the rework fraction increases after the temporary problem is resolved due to the higher schedule pressure. This activates loop R2, increasing the addition of new tasks, project backlog, and schedule pressure. Eventually, project conditions exceed the tipping point again (pt. 4), and the project crosses the tipping point a second time, this time due to endogenous causes. The evolution of percent complete for this project is shown in Figure 13.

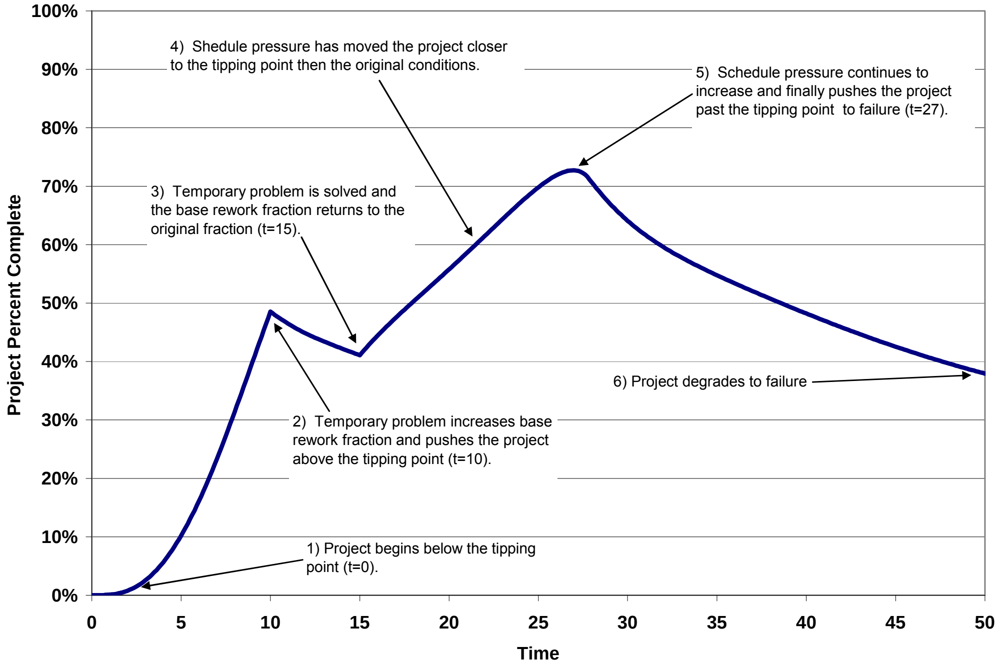

Figure 13. Project Trajectory Reversal Percent Complete

The behavior pattern in Figure 13[11] is similar to the behavior of the TVA's Watts Bar unit 2 project shown in Figure 1. This shows that the combination of a tipping point structure and ripple effects can cause projects to fail. A project experiencing this type of behavior (ever-increasing backlogs and decreasing percent complete) would be faced with making major changes (i.e., increasing resources, scope reduction, revising project deadline) or terminating the project. Either way, the project would likely be considered a failure (increasing costs, lost revenues due to delays) and negatively impact all involved entities.

We have demonstrated several scenarios in which an improving project can experience trajectory reversal and degrade to failure. Schedule pressure, an endogenous influence, can also push a project to failure. The experience shown in Figure 12 reflects a common problem that can be generated by a simple feedback structure. In the next section, we use this problem as the basis for testing strategies for managing development projects near a tipping point to avoid project failure or save projects that are degrading.

Project Management near Tipping Points

A review of current literature reveals several strategies for addressing tipping point failure in single development projects. These strategies for tipping point avoidance can be

[11] As defined for **Figure 13**, the Project Percent Complete is the work released as a fraction of the sum of the scope and added work. The Project Percent Complete can increase if projects are just slightly beyond the tipping point and if a large fraction of the new tasks added to the project backlog are simultaneously being approved and released. This can be shown by disaggregating the project backlog into the scope, total backlog added, and added backlog that is completed.

divided into avoiding tipping points through project design, resource management, backlog management, and schedule management.

Avoiding Tipping Points through Project Design

Avoiding tipping point conditions such as those described above entails the selection or design of development projects with relatively low rework fractions and added new tasks. Project selection may include an assessment of project complexity and interdependence and their impacts on the probability of success. Some projects or portions of projects may have characteristics that allow this strategy. For example, construction projects often use relatively simple technologies and processes to constrain rework fractions, and project planning purposefully keeps these operations separate to constrain ripple effects. Even projects with inherently high rework fractions and added new tasks can be designed to apply this strategy through methods such as modular design (Baldwin & Clark, 2000). Modular design develops projects as sub-systems that can be adjusted independently with minimal impact on the design as a whole. By designing projects with loose dependencies, if a design change does arise, it can be corrected with minimal impact on other systems. Figure 6 helps illustrate how a modular project with relatively low add-new-task strength would be able to tolerate a higher rework fraction without crossing the tipping point (i.e., it would reside in the lower-left of the chart). In the model, modular designed projects would have a lower add-new-task strength and would, therefore, be insulted from additional work added by the ripple effect. Modular design allows complex projects (with high rework fractions) to progress because a reduction in added new tasks has been designed into the project.

An example of modular design from the automotive industry is Toyota's method for designing components of a new car model. During the design of a new model, Toyota will provide their brake-system supplier with specifications regarding the weight of the car, the desired stopping distance from a given speed, and how much space the brake system can occupy in the wheel assembly (Womack et al., 1991). The brake supplier can change the design of the brake system without impacting other project components as long as the required specifications are met. This example demonstrates the concept of robustness which we apply to project design.

Robustness in Project Design

Taguchi et al. (2000) defines robustness as "the state where the product/process design is minimally sensitive to factors causing variability." The research of robustness in new product development has been largely limited to the robustness of the final product (Lou et. al., 2005; Swan et al., 2005). The current work expands the concept of robustness to project design and measures the protection that the robustness of a project provides from tipping point failure. An inspection of equation (5) suggests that, if a project starts far enough away from its tipping point (i.e. $f_{rw}(s_{nt}+1)<<1$) and increases in the rework fraction and the addition of new tasks strength are small, the project will not cross the tipping point and will monotonically improve. However, if the magnitude of the changes is large enough, the project could be pushed past the tipping point. By modeling robustness (r_{tp}) as the distance between project conditions and the tipping point, equation (5) can be rearranged to provide an intuitive meaning of project robustness against tipping point-induced failure:

$$f_{rw} + (f_{rw} * s_{nt}) + r_{tp} = 1 \qquad (6)$$

Where:

r_{tp} – project robustness to tipping point-induced failure {dimensionless}

The right side of equation (6) represents 100% of the project's capacity to tolerate additional new tasks. This capacity has been disaggregated into the three parts on the left side of equation (6): 1) capacity fraction absorbed by rework (f_{rw}), 2) capacity fraction absorbed by addition of new tasks ($f_{rw} * s_{nt}$), and 3) the unutilized capacity fraction that provides robustness (r_{tp}).When r_{tp} is positive, the project is below the tipping point (improving); when it is zero, the project is at the tipping point (stagnant); and when it is negative, the project is above the tipping point (degrading). For example, suppose a project has a fixed 20% reference rework fraction (f_{rw-r} = 0.2) and a fixed add-new-tasks strength (s_{nt}) of 1. Applying equation (6), this project begins 0.6 from the tipping point (has an initial robustness of 60%). Given these conditions, the project could tolerate schedule pressure-driven increases in the rework fraction of up to 30% (making f_{rw} = 50%) without crossing the tipping point.

Equation (6) also provides a means of analyzing the effects of different variables on project robustness. Robustness can vary significantly from initial conditions during a project. For example, schedule pressure can increase the fraction of work requiring change (f_{rw}) and, thereby, reduce robustness (equation 6). The minimum distance that project conditions come to the tipping point during the project represents a project's most vulnerable conditions. Therefore, a project's minimum distance from a tipping point is a better measure of project robustness than the initial distance. Figure 14 shows the results of a sensitivity analysis of project robustness to five variables that impact tipping point dynamics in the model.

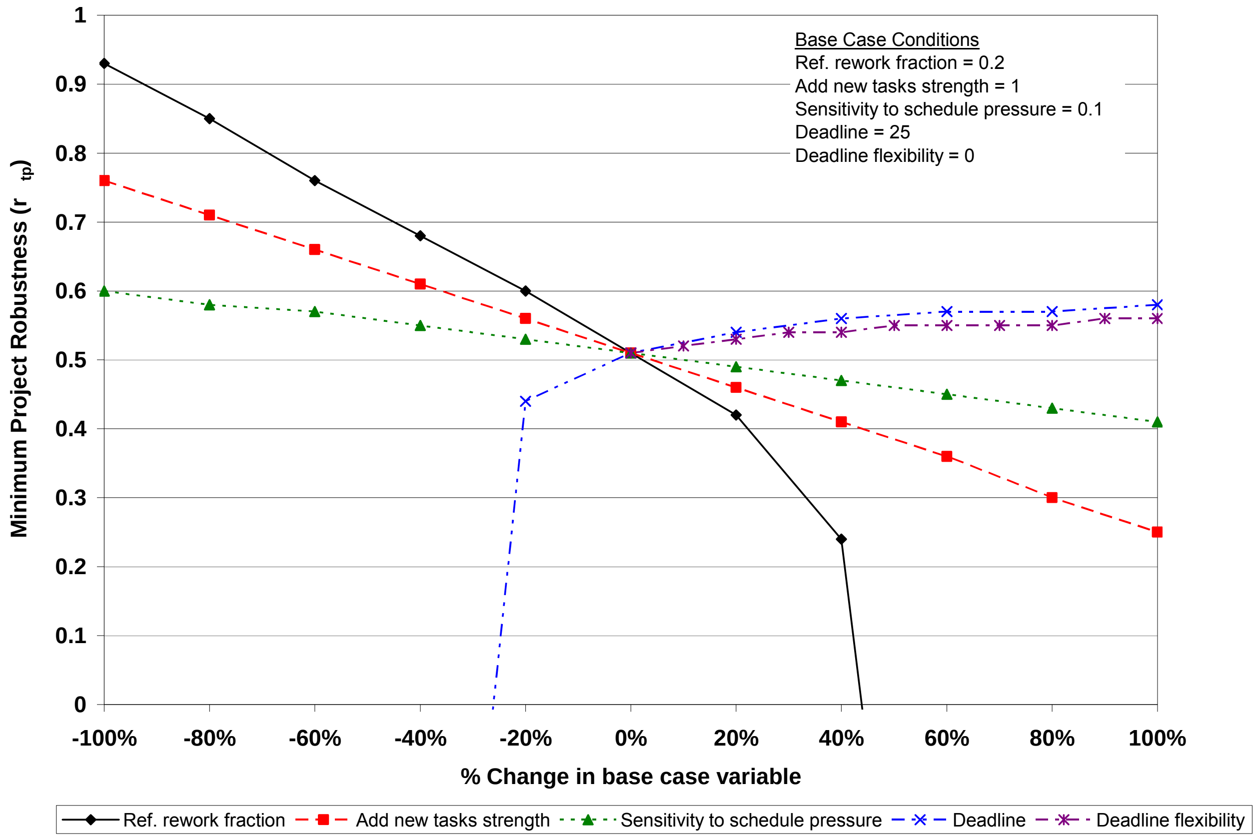

Figure 14. Sensitivity Analysis Results

The horizontal axis of Figure 14 represents the percent change from base-case values of the reference rework fraction, add-new-tasks strength, rework sensitivity to schedule pressure, deadline without flexibility, and flexibility of deadline. The vertical axis represents the project robustness or protection from tipping point-induced failure. For the base case, the robustness at the beginning of the project (60%) is reduced by schedule pressure during the project to a minimum of 51%. Values which "fall off" the bottom of the chart reflect negative robustness, when the project has crossed the tipping point and failed. The sensitivity analysis reveals two important features of the relationships between the control variables and minimum project robustness against tipping point-induced failure. First, with the exception of deadline flexibility, each variable has a threshold value, beyond which robustness quickly becomes negative. The threshold values for minimum robustness sensitivity to schedule pressure and add-new-tasks strength are 250% and 120% of the base-case conditions, respectively (not shown for clarity). In this analysis, deadline flexibility does not have a threshold value because the base-case project succeeds with no deadline flexibility. Therefore, adding flexibility cannot degrade performance. Second, within the robust ranges, the control levers vary in their impacts on robustness. By inspection of Figure 14, minimum project robustness is most sensitive to the reference rework fraction, then add-new-tasks strength, then rework sensitivity to schedule pressure, then deadline (inflexible), and is least sensitive to deadline flexibility.

Resource Management

Resource management includes altering the quantities of resources, their productivities, altering resource priorities to meet resource demands, anticipating future resource demands, and adjusting resources from current to needed applications. One reasonable response to a project that has crossed the tipping point is to add more resources to the project. The justification would be that since a project has more work, more resources are needed to complete the work. Model simulations show that increasing a project's resource level when a project crosses the tipping point can "save" the project, but this must be approached carefully. If adding resources does not reduce the rework fraction adequately through increased expertise, (for example) reduced schedule pressure, or other factors, the tipping point dynamics remain effectively the same. This can often be the case. Brooks (1982) states that, "adding manpower to a late software project makes it later." Likewise, if inexperienced resources are added to a project, particularly one that is complex, the amount of discovered rework could increase (Graham, 2000; Lyneis et al., 2001). In these cases, adding resources to a project that has begun to degrade would *increase* the rate of degradation. More resources making more mistakes would drive the project beyond the tipping point faster than fewer resources. Therefore, managers must be careful when adding resources to a project near tipping points.

Often the preceding strategy is unavailable because resource quantities for development projects are limited or fixed. A second strategy is to allocate resources to maximize the flows of work through the project. Certain backlogs could be given priority to resources based upon a manager's understanding of the critical aspects of the system. Black and Repenning (2001) studied this policy in multi-project systems. Repenning (2001) argues that "creating 'fire-resistant [tipping point resistant]' [new product development] systems requires the development of more dynamic methods of resource planning." He suggests that this planning method use the present state of the system to forecast the future resource needs. The basic model as described above follows this recommendation. In the basic model, managers are assumed to allocate the same fraction of resources to each activity as the activity's *current* backlog contributes to the project backlog.

A simple and reasonable extension of this policy is to assume that managers base allocations on their forecasts of resource needs at a time in the future. This is consistent with Cooper's (1994) suggestion that "developing an information system to forecast resources committed to known projects as well as resource availability as a function of time is no easy task, but it is essential." Thomke and Fujimoto (2000) suggest shifting resources to earlier parts of projects as the key to success, stating "faster product development can be achieved with an earlier generation of problem-and-solution related information, particularly if it involves critical path activities." Joglekar and Ford (2005) use a control-theory model and system dynamics to evaluate the impacts of forecasting resource demand on project schedule performance. Sterman's (2000, p. 634-636) structure for modeling trends is adopted[12] and the resulting trend linearly extrapolated from current backlog sizes into the future, the time required to reallocate resources. Sterman (2000) describes and explains the model structure and the equations that govern the resource forecasting system.

---

[12] The exception to the use of Sterman's trend structure is that only two exponential smoothing loops (rather than three) are used. Sterman's structure uses the third exponential loop to smooth "noisy" data fed into the structure. The input data to our structure is already smooth, so this third loop is unnecessary.

Resource Management Impacts on a Degrading Project

We simulated the potentially successful project (that became trapped beyond the tipping point and failed) (Figure 12) across a range of resource adjustment times and demand forecasting policies from no forecasting to forecasting with long-time horizons. As shown in Figure 15, resource forecasting can save the project. The project begins below the tipping point (pt. 1). At some point, a temporary problem is encountered that pushes the project past the tipping point (pt. 2). Once this problem is resolved, the project returns below the tipping point. Schedule pressure pushes the project close to the tipping point (pt. 4) after the project recovers from the temporary problem. As RW and QA backlogs begin to increase, resources are shifted far enough in advance and fast enough to prevent the backlogs (through schedule pressure) from pushing a project across the tipping point. A policy that uses four weeks of backlog history to develop a trend that is projected four weeks into the future can save the project if adjusted quickly enough (≤ 4 weeks).

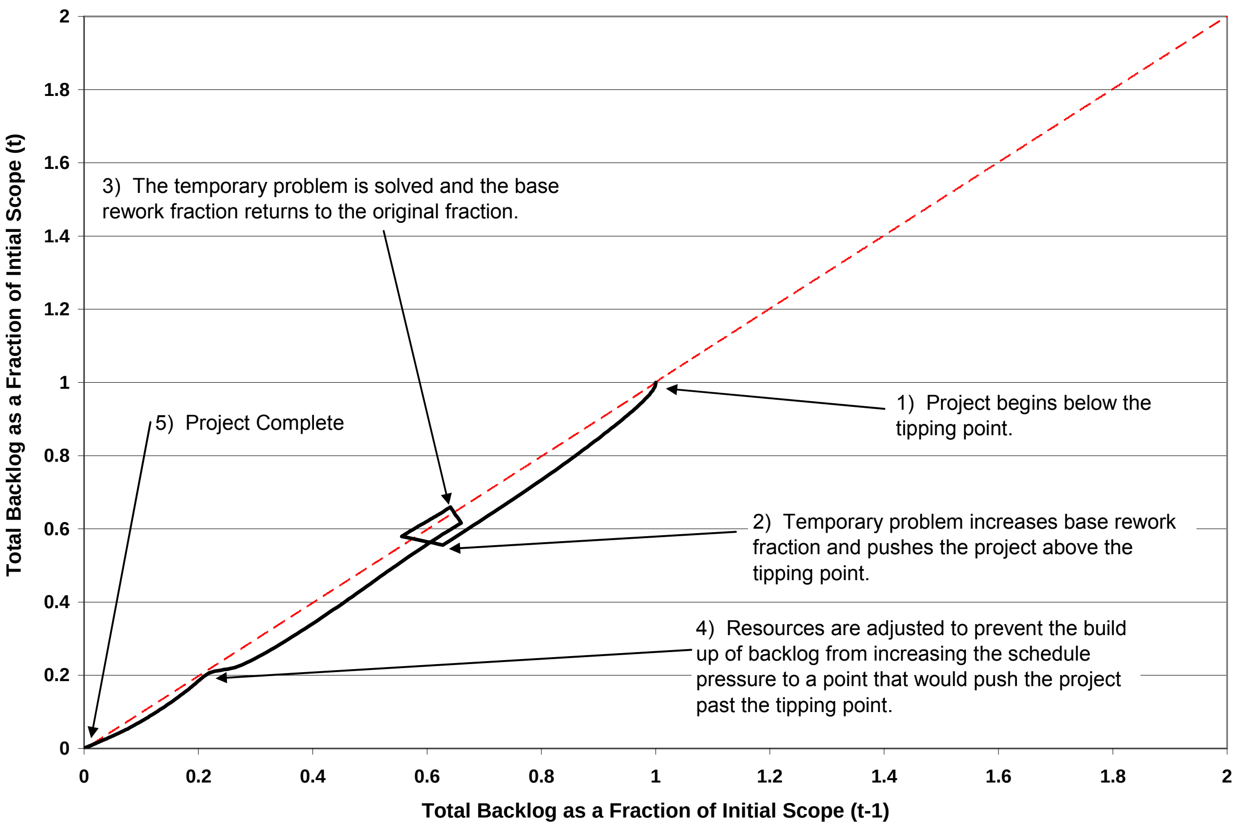

Figure 15. Use of Resource Forecasting to Save Project

Initial results show that longer trend adjustment times (i.e., a slow reacting manager) prevent the trend from reacting quickly enough to the increases in backlogs to allow resources to be allocated fast enough to save the project. Shorter trend adjustments (i.e., a quick reacting manager) pull the project farther away from the tipping point. This suggests that managers should react quickly to changes in project work backlogs.

Resource Adjustment Times

The time required to shift resources across development activities also impacts performance. Lee, Ford, and Joglekar (2004) found that resource adjustment times can have important impacts on project schedule performance. Their model simulations identified that there exists optimal resource adjustment times that minimize project duration over a range of project complexities. Reducing resource adjustment times while still utilizing proportional resource allocation policies can also save the project. Figure 16 shows the problem project (Figure 12) with a resource adjustment time reduced from 4 weeks to 3 weeks. Faster adjustments of resources towards target levels cause the schedule pressure to decrease after the removal of the temporary problem (Figure 16 pt. 4) and saves the project from continuous degradation (Figure 16).

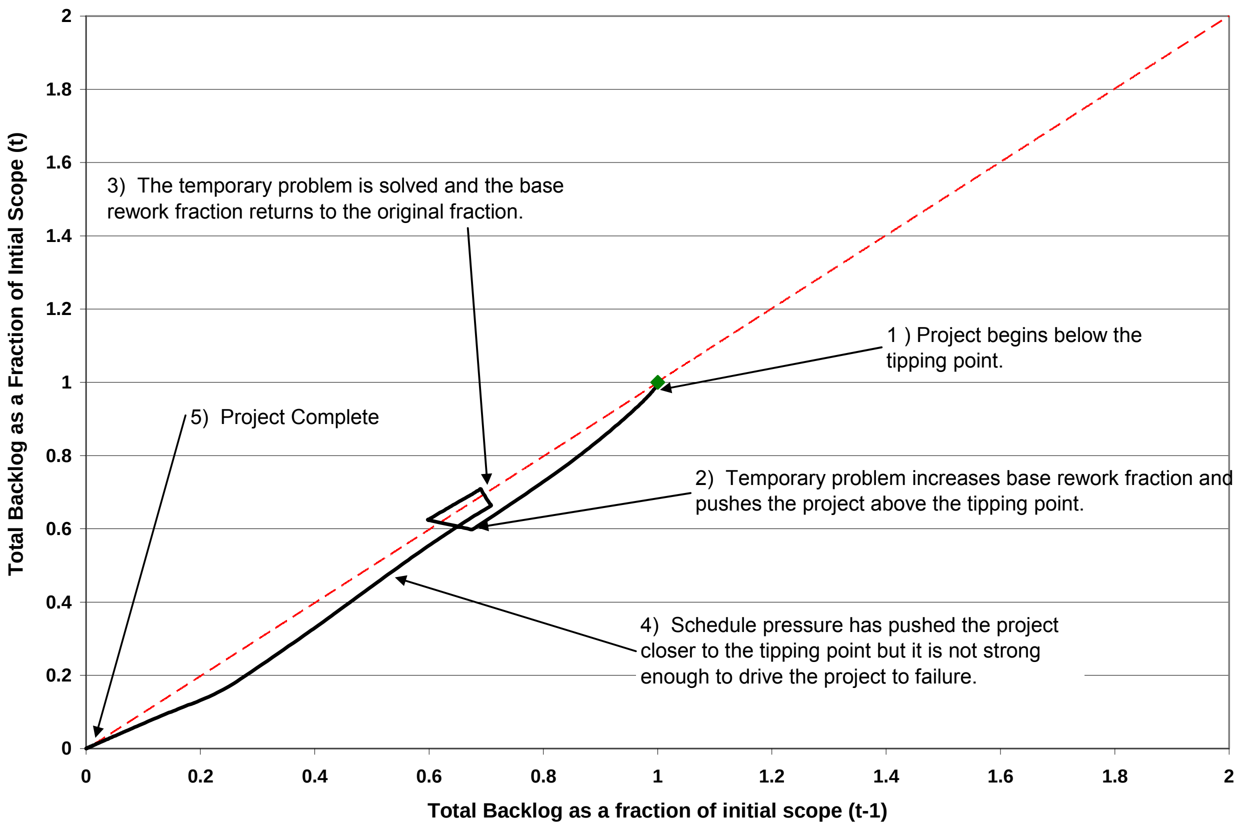

Figure 16. Use of Decreased Resource Adjustment Time to Save Project

Both resource forecasting and reduced staff adjustment times provide managers levers that can be used to save failing projects. Forecasting resources is somewhat straight forward, provided a manager can make a reasonable estimate of expected future work. From the authors' own experience the success of this policy is highly dependant on the accuracy of the estimate. One must also ensure that a change in the projected trend is reflective of changing resource requirements. Reducing staff adjustment times can be more challenging than resource forecasting, but appear to have a greater impact.

Backlog Management

Backlog management involves canceling work or releasing defective work. Work cancellation is a reduction during the project of features or scope of a project. Model simulations (not shown here for brevity) show that, if enough work is canceled, that canceling defective work can prevent a project from being overwhelmed with rework. As expected, projects nearer the tipping point required the cancellation of more work than those farther away. Black and Repenning (2001) found similar results using work cancellation in a multi-project system.

Schedule Management

One factor that controls schedule pressure is the project deadline. Both Cooper (1994) and Graham (2000) argue that setting realistic project deadlines reduces the amount of rework on a project. Therefore, an important part of schedule management is monitoring the project deadline and ensuring that it is realistic. One way to ensure a realistic deadline is to implement a flexible deadline. A flexible deadline is dependent upon the amount of work left to be completed in a project. A rigid deadline does not take into account changes or delays in a project caused by rework and added new tasks. Flexible deadlines take these effects into account by adjusting the expected completion date based upon the time required to complete the total project backlog. To model this flexibility, the project deadline moves toward the expected completion date at a rate based on the flexibility of the deadline and the difference between the expected completion data and deadline. The effectiveness of flexible deadlines in saving the problem project (Figure 12) was tested. Figure 17 shows how a flexible deadline prevents the increased backlog due to the temporary problem from increasing schedule pressure. This prevents schedule pressure from building up to a point that would drive the rework fraction high enough to push the project beyond the tipping point. This suggests that managers can use deadline flexibility to recover projects from degradation initiated by crossing a tipping point.

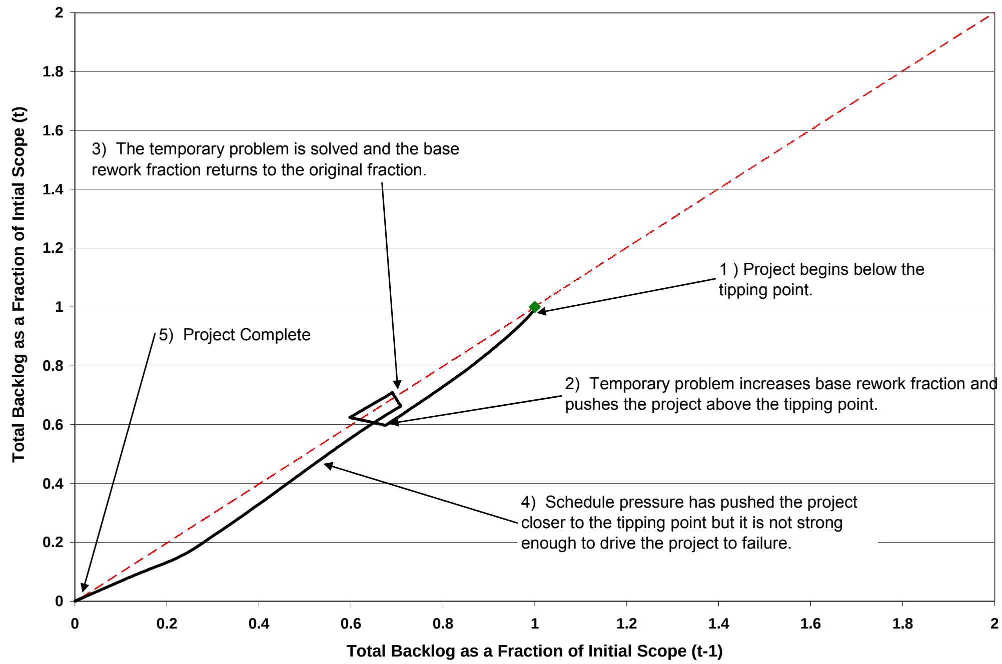

Figure 17. Use of Flexible Schedule on Project Past the Tipping Point

Tipping-point Management in the Nuclear Plant Construction Industry

Unlike Watts Bar #2, Limerick Unit #2 was ultimately completed. A review of the methods used to complete the unit reveals the use of several of the tipping point management policies previously discussed. Construction of Limerick Unit #2 was resumed in February of 1986, and the unit was completed by August of 1989 by Bechtel. To complete the unit, Bechtel implemented several of the solutions previously presented. The overall backlog to be completed was reduced by eliminating many of the required pipe supports through an advanced support design (Clarey, 1987) illustrating the use of backlog management. Bechtel also increased the non-manual manpower on the project by 300% a year before increasing the manual workforce. Nearly two-thirds of this increase was in the form of engineering and construction management personnel (Clarey, 1987) which improved the project's work planning. Once manual work began, all complex installations were thoroughly reviewed by engineers to reduce overall rework, and design engineers were placed on all working shifts to resolve any constructability issues that arose (Clarey, 1987). This illustrates the concept of increasing manpower in a way that potentially reduces the rework and ripple effects of the project.

Conclusions

Tipping point structures are integrated with single development project dynamics to examine project behavior modes. The tipping point is useful because it allows comparison of project failure due to different causes. Rework and the addition of new tasks can combine to push projects to fail. By understanding these failure mechanisms, potentially robust policies are examined that can decrease the risk of failure for projects near the tipping point threshold. Successful policies were those that avoided the tipping point by reducing rework and ripple effect or those that reduced backlogs by effectively managing resources.

The policies tested provide several managerial implications. Tipping point conditions (Eq. (4)) support the use of modular design in the development of complex products. By reducing the ripple effect, modular design would allow more aggressive projects to be pursued with reduced risk of failure. As described in the discussion of Toyota's brake system design, modular design allows concurrent development of project tasks with minimal interdependence. Project managers would benefit from preliminary designs which set project specifications to allowing concurrent modular design. The work also contributes a preliminary test of robustness as a measure of future project performance. Our results show that robustness may be a good measure of a project's protection from tipping point failure. Future research in this area should focus on operationalizing robustness for use across a wide range of project types. This future work could provide project managers with a method of evaluating the failure potential of projects.

Proper resource management can play an important role in project success. Resource forecasting (with quick identification of changing trends) has the potential to further insulate projects from the tipping point. Model simulations show that the most successful policies are those which are short in hindsight and forecast farther into the future. However, one limit of the model used here is that it benefits from data free of the "noise" typically associated with actual project tracking reports. Managers must be careful to ensure that a perceived project trend change is an actual change in project progression and not normal oscillations in project progress reporting before adjusting resources.

Resource adjustment times were also found to be potentially effective in responding to projects vulnerable to tipping point dynamics. Quicker adjustment times for both proportional and forecasted resource allocation policies were beneficial in preventing projects subject to schedule pressure from crossing the tipping point. Again, the model is limited in that it does not take into account the negative effects (worker morale, lost production time, etc.) of shifting resources. Other work (Lee et al., 2004) has shown that there is an optimal adjustment time for resources, remaining below which can be detrimental to a project. While flexible resources can be beneficial to a project, the key for managers is to ensure that resources are be adjusted in an efficient manner.

Finally, realistic deadlines can help prevent a project from being overwhelmed with schedule pressure. Managers need to carefully consider how changes in work volume, through either increased rework or scope changes, affect a project's deadline. Model simulations show that managers should resist the temptation to strive for schedules which have become unrealistic due to drastic changes in a project's work volume. This is supported by other research (Lyneis et al., 2001; Graham, 2000).

The model structure used in this work has several limitations. This includes the assumption that all work released is of one quality. This prevents a policy investigation similar to Black and Repenning (2001) of releasing lower quality work in a single project system. In addition, the model does not take into account work that must be redone due to rework. Improved models that take into account these conditions are needed to fully examine polices that govern single project development. Future research can improve model structure consistency with actual projects and calibrate the model to practice.

Tipping point dynamics can strongly influence the behavior and performance of individual development projects, and sometimes determine their success or failure. Continued improvement in the understanding of tipping point dynamics can lead to better development, project management and performance.

Acknowledgement: The authors thank Prof. Nitin Joglekar for suggesting the positioning of this paper, Prof. Kenneth Reinschmidt for assisting in nuclear power plant investigation, and Marsha Ward, NRC Research Librarian, for assisting in the acquisition of NRC nuclear power plant construction data and three anonymous reviewers of a draft of this paper.

References

Adbel-Hamid, T.K. (1984). The dynamics of software development project management: An integrative systems dynamics perspective. (Doctoral dissertation). Cambridge, MA: Massachusetts Institute of Technology.

Arditi, D. & Kirsininkas, A. (1985). Nuclear power plant delays: Probable causes and resolutions. *Transition in the Nuclear Industry*. New York: American Society of Civil Engineers.

Aron, J. (1997). *Licensed to kill? The nuclear regulatory commission and the Shoreham power plant*. Pittsburgh, PA: University of Pittsburgh Press.

Baldwin, C., & Clark, C. (2000). *Design rules: The power of modularity*. Cambridge, MA: MIT Press.

Black, L., & Repenning, N. (2001). Why firefighting is never enough: preserving high-quality product development. *System Dynamics Review,* 17(1), 33-62.

Brooks, F. (1982). *The mythical man-month*. Philippines: Addison-Wesley Publishing Company.

Busby, S., & Hughes, E. (2004). Projects, pathogens, and incubation periods. *International Journal of Project Management*, 22(2), 425-434.

Clarney, J. (1987, November). New concept cuts nuclear project construction cost. *Power Engineering,* 28-31.

Cooper, K.G. (1980). Naval ship production: A claim settled and a framework built. *Interfaces*, 10(6), 20-36.

Cooper, K.G. (1993a, February). The rework cycle: Why projects are mismanaged. *PM network,* 5-7.

Cooper, K.G. (1993b, February). The rework cycle: How it really works…and reworks. *PM network,* 25-28.

Cooper, K.G. (1993c). The rework cycle: Benchmarks for the project managers. *Project Management Journal,* 14(1), 17-21.

Copper, K.G. (1994). The $2,000 hour: How managers influence project performance through the rework cycle. *Project Management Journal,* 15(1), 11-24.

Cooper, K., Lyneis, J., & Bryant, B. (2002). Learning to learn, from past to future. *International Journal of Project Management,* 20(3), 213-219.

Days, M., & Sellers, V. (1985, October 18). Philadelphia firm allowed to finish nuclear plant. *The Wall Street Journal*, p. 1.

Evans, M. (2005). Overdue and over budget, over and over again; Project management. *The Economists,* 375(8430), 66.

Feldman, S., Bernstein, M., & Noland, R. (1988). The costs of completing unfinished US nuclear power plants. *Energy Policy*, 16(3), 270-279.

Ford, D. (1995). The dynamics of project management: An investigation of the impacts of project process and coordination on performance. (Doctoral Dissertation). Cambridge, MA: Massachusetts Institute of Technology.

Ford, D., & Sterman, J. (1998). Modeling dynamic development processes. *System Dynamics Review*, 14(1), 31-68.

Ford, D., & Sterman, J. (2003a). The liar's club: Impacts of concealment in concurrent development projects. *Concurrent Engineering Research and Applications,* 111(3), 211-219.

Ford, D., & Sterman, J. (2003b). Overcoming the 90% syndrome: Iteration management in concurrent development projects. *Concurrent Engineering Research and Applications,* 111(3), 177-186.

Friedrich, D., Daly, J., & Dick, W. (1987). Revisions, repairs, and rework on large projects. *Journal of Construction Engineering and Management*, 113(3), 488-500.

Gladwell, M. (2000). *The tipping point: How little things make a big difference*. Boston, MA: Little, Brown, and Company.

Graham, A. (2000). Beyond PM 101: Lessons for managing large development programs. *Project Management Journal,* 31(4), 7-18.

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

Kharbanda, O., & Pinto, J. (1996). *What made Gertie Gallop: Learning from project failures*. New York: Van Nostrand Reinhold.

Kim, D.H. (1988). Sun Microsystems, Sun3 product development/release model. Technical Report D-4113. Cambridge, MA: Systems Dynamics Group, Sloan School of Management, Massachusetts Institute of Technology.

Lake, J. (2002). The fourth generation of nuclear power. *Progress in Nuclear Energy,* 40(3-4), 301-307.

Lee, N. (1995, December 3). Powering up of TVA reactor generates safety debate. *The Washington Post,* pp. A22-A-23.

Lee, Z, Ford, D., & Joglekar, N. (2004). The design of resource allocation policies. *Proceedings of the 2004 International System Dynamics Conference.* Oxford, UK. July 25-29.

Lillington, J. (2004). *The future of nuclear power*. UK: Elsevier.

Luo, L., Kannan, P., Besharati, B., & Azram, S. (2005). Design of robust new products under variability: Marketing meets design. *Journal of Product Innovation Management*, 22(2005), 177-192.

Lyneis, F., Cooper, K., & Els, S. (2001). Strategic management of complex projects: A case study using system dynamics. *Systems Dynamics Review*, 17(3), 237-260.

Matta, F., & Ashkenas, R. (2003, September). Why good projects fail anyway. *Harvard Business Review,* 109-114.

Nuclear Engineering International. (1995, February). TVA announces the end of the line for Watts Bar 2, Bellefonte 1 & 2.

Nuclear Regulatory Commission (NRC). (1982, June 30). *Nuclear power plant construction status report 06/30/82*. NUREG-0030 6(2).

Paul, L. (2005). It’s your move. *PM Network,* 19(4), 34-38.

Reinschmidt, K. (2005, March 11). [Personal Interview with author]. College Station, TX.

Repenning, N. (2001). Understanding fire fighting in new product development. *Journal of Product Innovation Management*,18(2001), 265-300.

Richardson, G.P., & Pugh, A.L. (1981). *Introduction to systems dynamics modeling with dynamo*. Cambridge, MA: MIT Press.

Rodrigues, A. & Williams, T.M. (1997). System dynamics in project management: assessing the impacts of client behavior on project performance. *Journal of the Operations Research Society*, 49, 2-15.

Sterman, J.S. (2000). *Business dynamics: Systems thinking and modeling for a complex world*. New York: Irwin McGraw-Hill.

Swan, K., Kotabe, M., & Allred, B. (2005). Exploring robust design capabilities, their role in creating global products, and their relationship to firm performance. *Journal of Product Innovation and Management,* 22(1), 144-164.

Taguchi, G., Chowdhury, S., & Taguchi, S. (2000). *Robust engineering*. New York: McGraw-Hill.

Taylor, T, Ford, D., & Johnson, S. (2005). When good projects go bad: Tipping point dynamics in development projects. *2005 International System Dynamics Conference*, Boston, MA. July 17-21, 2005.

Taylor, T., & Ford, D. (2006). Tipping point failure and robustness in single development projects. *System Dynamics Review*. In press.

Thomas, H. R., & Napolitan, C. L. (1994, August). The effects of changes on labor productivity: Why and how much. SD No. 99. Austin, TX: Construction Industry Institute.

Thomke, S., & Fujimoto, T. (2000). The effect of "front-loading" problem solving on product development performance. *Journal of Product Innovation Management*, 17(2000), 128-142.

United States General Accounting Office (USGAO). (1996). *Department of Energy: Opportunity to improve management of major systems acquisitions*. GAO/RCED-97-17. Washington, DC: author.

United States General Accounting Office (USGAO). (1997). *Department of Energy: Improving management of major systems acquisitions*. GAO/T-RCED-97-92. Washington, DC: author.

Wells, W. (1999). From the editor. *Project Management Journal,* 30(2), 4-5.

Womack, J. Jones, D., & Roos, D. (1991). *The machine that changed the world*. New York: Harper Perennial.

THIS PAGE INTENTIONALLY LEFT BLANK

2003 - 2006 Sponsored Acquisition Research Topics

Acquisition Management

- Software Requirements for OA
- Managing Services Supply Chain
- Acquiring Combat Capability via Public-Private Partnerships (PPPs)
- Knowledge Value Added (KVA) + Real Options (RO) Applied to Shipyard Planning Processes
- Portfolio Optimization via KVA + RO
- MOSA Contracting Implications
- Strategy for Defense Acquisition Research
- Spiral Development
- BCA: Contractor vs. Organic Growth

Contract Management

- USAF IT Commodity Council
- Contractors in 21st Century Combat Zone
- Joint Contingency Contracting
- Navy Contract Writing Guide
- Commodity Sourcing Strategies
- Past Performance in Source Selection
- USMC Contingency Contracting
- Transforming DoD Contract Closeout
- Model for Optimizing Contingency Contracting Planning and Execution

Financial Management

- PPPs and Government Financing
- Energy Saving Contracts/DoD Mobile Assets
- Capital Budgeting for DoD
- Financing DoD Budget via PPPs
- ROI of Information Warfare Systems
- Acquisitions via leasing: MPS case
- Special Termination Liability in MDAPs

Logistics Management

- R-TOC Aegis Microwave Power Tubes

- Privatization-NOSL/NAWCI
- Army LOG MOD
- PBL (4)
- Contractors Supporting Military Operations
- RFID (4)
- Strategic Sourcing
- ASDS Product Support Analysis
- Analysis of LAV Depot Maintenance
- Diffusion/Variability on Vendor Performance Evaluation
- Optimizing CIWS Life Cycle Support (LCS)

Program Management

- Building Collaborative Capacity
- Knowledge, Responsibilities and Decision Rights in MDAPs
- KVA Applied to Aegis and SSDS
- Business Process Reengineering (BPR) for LCS Mission Module Acquisition
- Terminating Your Own Program
- Collaborative IT Tools Leveraging Competence

A complete listing and electronic copies of published research within the Acquisition Research Program are available on our website: www.acquisitionresearch.org

ACQUISITION RESEARCH PROGRAM
GRADUATE SCHOOL OF BUSINESS & PUBLIC POLICY
NAVAL POSTGRADUATE SCHOOL
555 DYER ROAD, INGERSOLL HALL
MONTEREY, CALIFORNIA 93943

www.acquisitionresearch.org