Project Controls to Minimize Cost and Schedule Overruns: A Model, Research Agenda, and Initial Results

David N. Ford[1], James M. Lyneis[2], and Timothy R.B. Taylor[3]

Abstract

System dynamics has been successfully applied to the study of projects for many years. While this modeling has clearly defined the structures which create project dynamics, it has been less helpful in providing explicit policy advice to managers to control project performance. To address this gap we examine the impact of three common project controls available to project managers to address deviations in project performance; (1) exerting pressure on project staff to work faster, (2) having staff work overtime, and/or (3) hiring additional staff. While the three project controls can have short-term benefits for project performance their long-term impacts can be detrimental. The current work presents preliminary results of the research, focusing on the impacts of the three project controls on project rework and the resulting schedule and budget performance. The work describes the development of project control feedback structures, the initial testing and use of a formal system dynamics model of the system, and preliminary results. The work concludes with a description of future project research efforts.

Introduction

Development projects are notorious for over-running their cost and schedule budgets (Flyvbjerg et al. 2003; Matta and Ashkenas 2003; Evans 2005; Nassar et al. 2005). Many factors can contribute to poor project cost and schedule performance including definition and selection, pre-project planning, and project execution. For example, in project planning infeasible deadlines or budgets or exogenous changes which increase the work required to fulfill requirements can prevent good performance. While proper project selection and planning can address many downstream issues that can cause poor project performance, managers often have to address problems that arise during project execution. During project execution managers can use several project controls to address poor project performance. As used here project controls are managerial decisions and actions intended to correct poor project performance during project execution. Project controls can include process improvements, adjusting performance targets, change management, and resource management. The current work focuses on resource management, specifically labor resource management, because labor project controls can cause many significant unintended negative side effects.

Labor resource management project controls seek to improve project performance by increasing the quantity of labor applied or the use of resources. For example, if a project is behind schedule, management can take some combination of (1) exerting pressure on project staff to work faster,

[1] Associate Professor, Construction Engineering and Management Program, Zachry Department of Civil Engineering, Texas A&M University, College Station, TX 77843-3136. (979) 845-3759 davidford@tamu.edu

[2] Professor of Practice, Department of Social Science and Policy Studies, Worcester Polytechnic Institute, 100 Institute Road, Worcester, MA 01609-2280. jmlyneis@wpi.edu.

[3] PhD Candidate, Construction Engineering and Management Program, Zachry Department of Civil Engineering, Texas A&M University, College Station, TX 77843-3136. xftu@tamu.edu

(2) having staff work overtime, and/or (3) hiring additional staff in an effort to get the project back on track. While these actions all help to increase the rate of progress in the short-term by increasing labor quantity, they also have both short- and long-term negative consequences for projects. Examples include 1) working faster increases errors ("haste-makes-waste"); 2) excessive overtime increases fatigue, ultimately reducing productivity and increasing errors; and 3) hiring additional staff can reduce short-term productivity as new staff are trained and gain experience, and can reduce long-term productivity due to increased organizational size and inefficiencies. How can project managers balance these costs and benefits to bring an underperforming project back on track?

System dynamics has an established history of modeling unintended side effect of managerial decisions, particularly in the project management field (see Lyneis and Ford 2007 for a survey of this work). While this modeling has clearly defined the structures which create project dynamics, it has been less helpful in providing explicit policy advice to managers. For example, what combination of the above resource project controls – that is, in what amounts, durations, and orders of application – will bring a project back on schedule with the least adverse consequences? How effective are commonly used heuristics? Can we provide better heuristics that will improve project performance, under a variety of different conditions?

The research described herein addresses these questions. The current paper lays out the research agenda, describes the model being developed to pursue the research, and presents preliminary results of model validation and use. First, we review project dynamics and the structures that create those dynamics. Then we present the model used in this research, followed by an analysis of the impacts of project control actions on project rework. We conclude with some preliminary implications and a discussion of directions for future research.

Project Dynamics

Typical project dynamics are illustrated in Figure 1. The project staffing plan builds up to a peak, and then gradually declines. But, more commonly, the actual ramp up of staff is delayed, then staffing levels overshoot the planned peak and extend longer (sometimes with a second peak). While the planned fraction complete grows linearly or with a slight s-shape, actual fraction complete is often below planned levels, and plateaus for a period of time before rising to completion. On some projects, fraction complete may cross a "tipping point" and begin to decline. Several system dynamics researchers have documented these and other consistent behavior modes in actual projects (e.g. Reichelt and Lyneis 1999; Love et al. 2000; Ford and Sterman 2003b; Taylor and Ford 2006). What causes these dynamics? In this section we discuss the contribution of different types of drivers to the basic theory of these project dynamics.

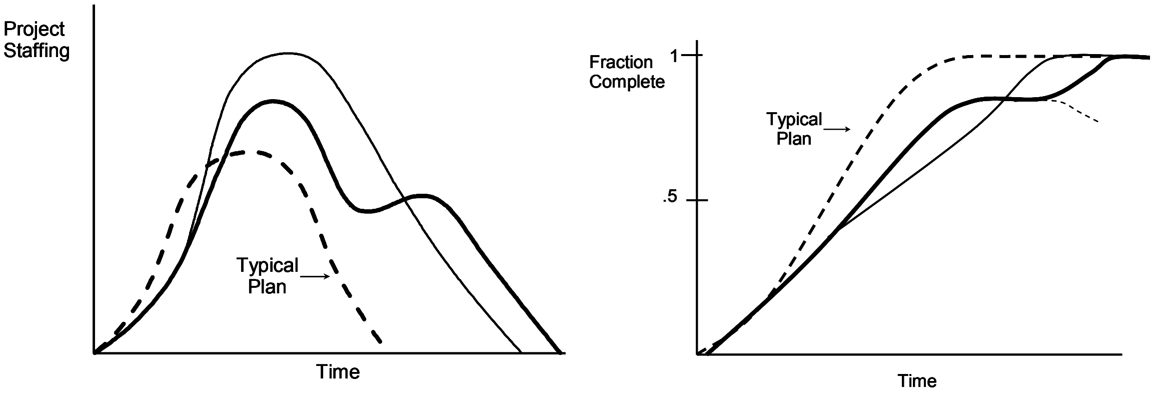

Figure 1: Typical project dynamics (adapted from Lyneis, et. al. 2001)

An underlying theory of the basic structures that create typical project behavior has emerged from system dynamics work on projects. As detailed in Lyneis and Ford (2007), four basic types of drivers generate these dynamics:

1. A **"rework cycle"** in which undiscovered errors in current work are later discovered and reworked, potentially cycling again through this process as errors are made in correcting rework;
2. **Controlling (negative or balancing) feedback loops** through which management attempts to bring a project which has fallen behind schedule, or over budget, back on track (e.g., adding staff, working overtime, exerting schedule pressure), or adjusts the targets (e.g. increasing deadlines or reducing scope);
3. **Ripple-effects**, the primary positive or re-enforcing feedbacks – vicious circles – which circumvent project control efforts (e.g., skill dilution from hiring staff; fatigue from overtime; "haste makes waste" from schedule pressure); and
4. **Secondary or "knock-on" effects**, within or between phases of work, caused by processes that produce excessive or detrimental concurrence of upstream and downstream (e.g., errors on errors feedback, unresolved problems in design create problems in construction), or human factors that amplify the negative effects via channels such as morale.

We next describe these structures in more detail.

Rework Cycle. At the heart of all project dynamics are the processes which describe the accomplishment of work, and as importantly, rework. Figure 2 shows the fundamental components and relationships of the rework cycle employed in this research. The basic structure is based on previous system dynamics research (Cooper 1993; Ford 1995; Ford and Sterman 1998; Taylor and Ford 2006). The rework cycle represents five stocks of work. At the start of a project or project phase all work resides in the stock "Initial Completion Backlog." As the project evolves, work is accomplished by applying effort (staff-hours) working at varying productivities, thereby depleting the pool of original work and increasing the two Quality Assurance (QA) backlogs. A certain fraction of the work being done at any point in time contains errors – that is, it is of less than perfect quality. Work that is done correctly enters the stock "QA Backlog – Correct Work," and requires no rework (unless later changes obsolete that work). However, work containing errors enters a stock of "QA Backlog – Incorrect Work."[4] Errors are not immediately recognized, but are detected as a result of doing QA work. Work completed incorrectly flows into the stock "Rework Backlog." The fraction of rework performed incorrectly is assumed to equal the fraction of initial work performed incorrectly. Correct rework increases the "QA Backlog – Correct Work"; incorrect rework increases "QA Backlog – Incorrect Work." As a result, tasks can make several iterations through the rework cycle.

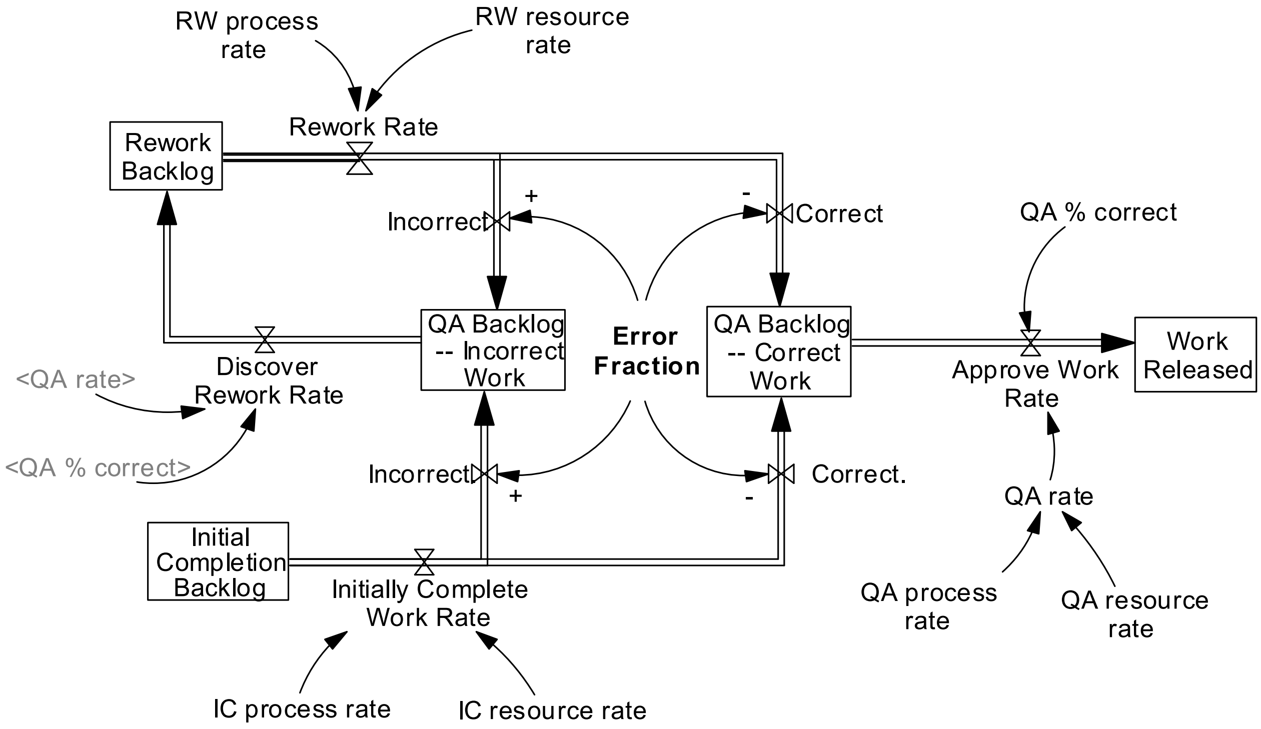

Figure 2: The rework cycle

Controlling Feedbacks. While the rework cycle in itself creates project dynamics as a result of the cycling of tasks containing errors, management actions to control a project to its cost, schedule, and quality objectives can also create errors. When actual progress suggests that performance at milestone events such as project completion will not meet objectives, actions are taken to bring the project back on line. There are two basic strategies, as illustrated in Figure 3: (1) improve performance by increasing or changing project effort, or (2) adjusting performance targets. Both strategies use negative feedback loops to close the gap between targets and forecasted performance. Examples of adjusting targets toward expected performance (right loop in Figure 3) include slipping deadlines, reducing scope, and the delivery of projects with defects.

[4] The QA backlog is modeled with two stocks even though project managers cannot distinguish between correct work and incorrect work in the QA backlog. This was done to reflect the generalization of rework impacts during initial completion and not during quality assurance. See Ford and Sterman (1998) for an alternative model using a co-flow structure to generate similar behavior.

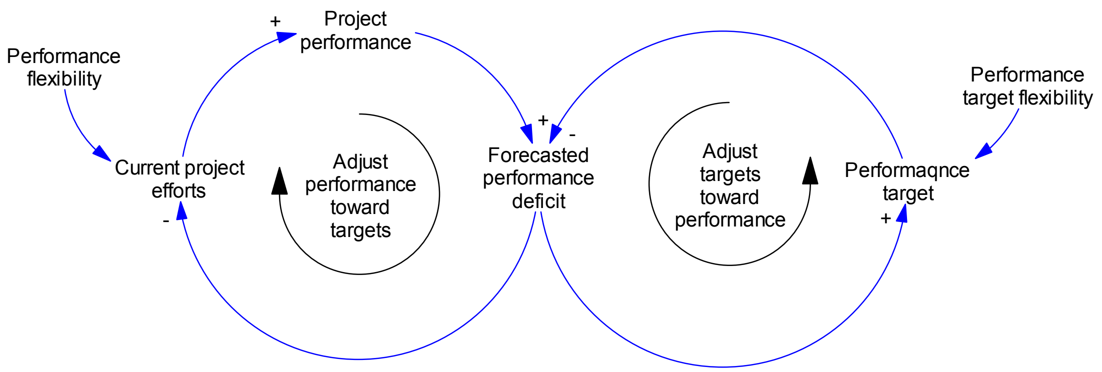
Performance flexibility
+
Project performance
Current project efforts
-
Adjust performance toward targets
+
-
Forecasted performance deficit
Adjust targets toward performance
Performance target flexibility
Performaqnce target
+

Figure 3: A generic structure of controlling feedback (assumes smaller is better)

While project performance targets can be adjusted, they are often inflexible. For example, "time to market" is often the primary project objective (e.g. to meet an annual release date, an announced product launch, to gain a first mover advantage, or the desire to beat competitors to market).[5] Therefore, typically the first (and often only) response to forecasted underperformance is to adjust project efforts (the left loop in Figure 3) to bring a project's progress back toward the targeted performance. Human self interest certainly encourages this bias for improving performance over shifting targets - nobody wants to admit they can't "get the job done on schedule and in budget." We use improving schedule performance in the current work to illustrate controlling feedback structures because schedule performance has been a particular focus of many system dynamics project models.

Why does a project fall behind schedule? There are many possible triggers: (1) the scope could have been underestimated or increased because of customer requests; (2) customers can request changes in the design or project requirements during the course of a project which cause completed work to require rework; (3) the original plan may have been infeasible (e.g. understaffed, too aggressively scheduled, or failing to take rework into account); or (4) other risks might have occurred, such as lack of staff due to hiring delays or delayed completion of upstream projects. Whatever the cause, resource management actions are often taken during the project (i.e. project controls) to correct the short-fall including working faster (increasing work intensity), working more hours (overtime), and adding people (increasing work force) as illustrated in Figure 4. These actions form negative feedback loops. If work remaining is more than can be completed in the time remaining labor resources needed exceed resources currently on hand. Overtime might then be increased, which increases the effective workforce (as the number of "full time equivalent" developers or FTEs) and effort applied on the project. With

[5] While actions to control a project toward the planned schedule are most common, other project objectives such as cost and delivered quality can also generate corrective actions. For example, if forecasted costs are above budget, either the scope can be reduced or the staffing levels cut back (especially if the budgets are annual budgets rather than project budgets or the cost of development funds is very high). Management can also slip the schedule. Because these alternative controlling actions are less common, we have focused our research on management policies to achieve the original schedule.

more effort, work progress increases, which causes the project backlog[6] (the dashed line in Figure 4) to decrease faster than they otherwise would have, thereby reducing work remaining faster than time remaining, and reducing, if not eliminating, the staff shortage. The "Work Faster" and "Add People" loops work similarly to this "Work Harder" loop, i.e. by increasing the effective workforce.

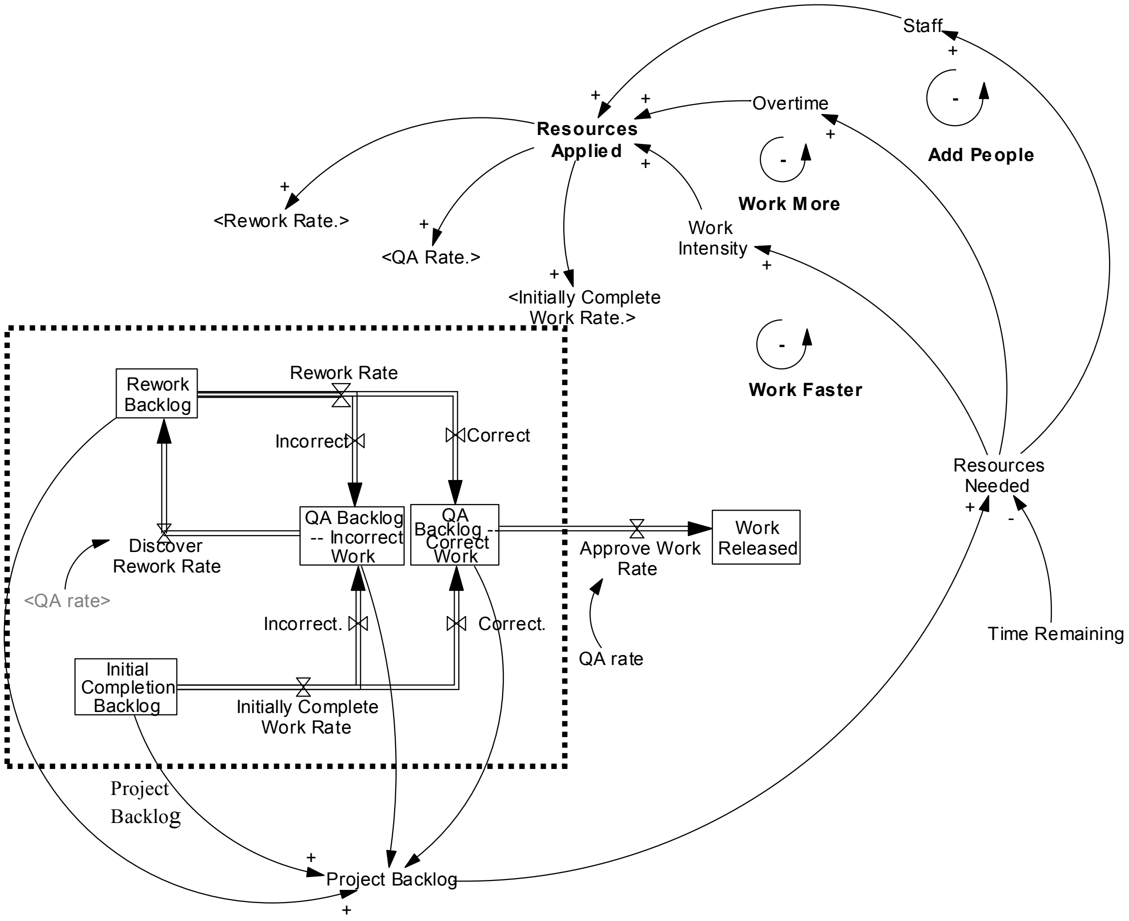

Figure 4: Controlling feedback loops for improving schedule performance

Ripple Effects – Ripple effects are the primary impacts of controlling feedback on rework and productivity. Unfortunately, actions taken to close a gap between project performance and targets have unintended side effects that generate policy resistance. As illustrated in Figure 5, which builds on the controlling feedbacks shown in Figure 4, these effects typically operate by increasing rework or reducing productivity. Each of the three primary forms of improving schedule performance (add staff, overtime, and press for greater output) can cause policy resistance. For example, adding staff can dilute experience as staff with less skill and/or less familiarity with the project are added and they require experienced developers to divert time to training instead of development. This reduces productivity and increases errors. This results in slower progress and increased rework. This increases work remaining, resulting in an increase

[6] The project backlog is the initial completion backlog + the rework backlog + the QA backlog.

staff demand. Additional staff is then added, further diluting experience. This cycle represents a positive feedback loop. The impact of increasing staff on rework is temporary as the new workers gain experience and the rework ripple effects decrease (Figure 6) Using overtime to improve schedule performance can (after a delay) cause developers to "burnout" due to fatigue from sustained overtime, increasing error fraction and reducing productivity (Figure 6). Pushing developers for more output can cause a "haste makes waste" dynamic which immediately increases errors produced on the project (Figure 6). All of these forms of policy resistance create positive feedback loops, or vicious circles, which can cause a project to spin out of control.[7]

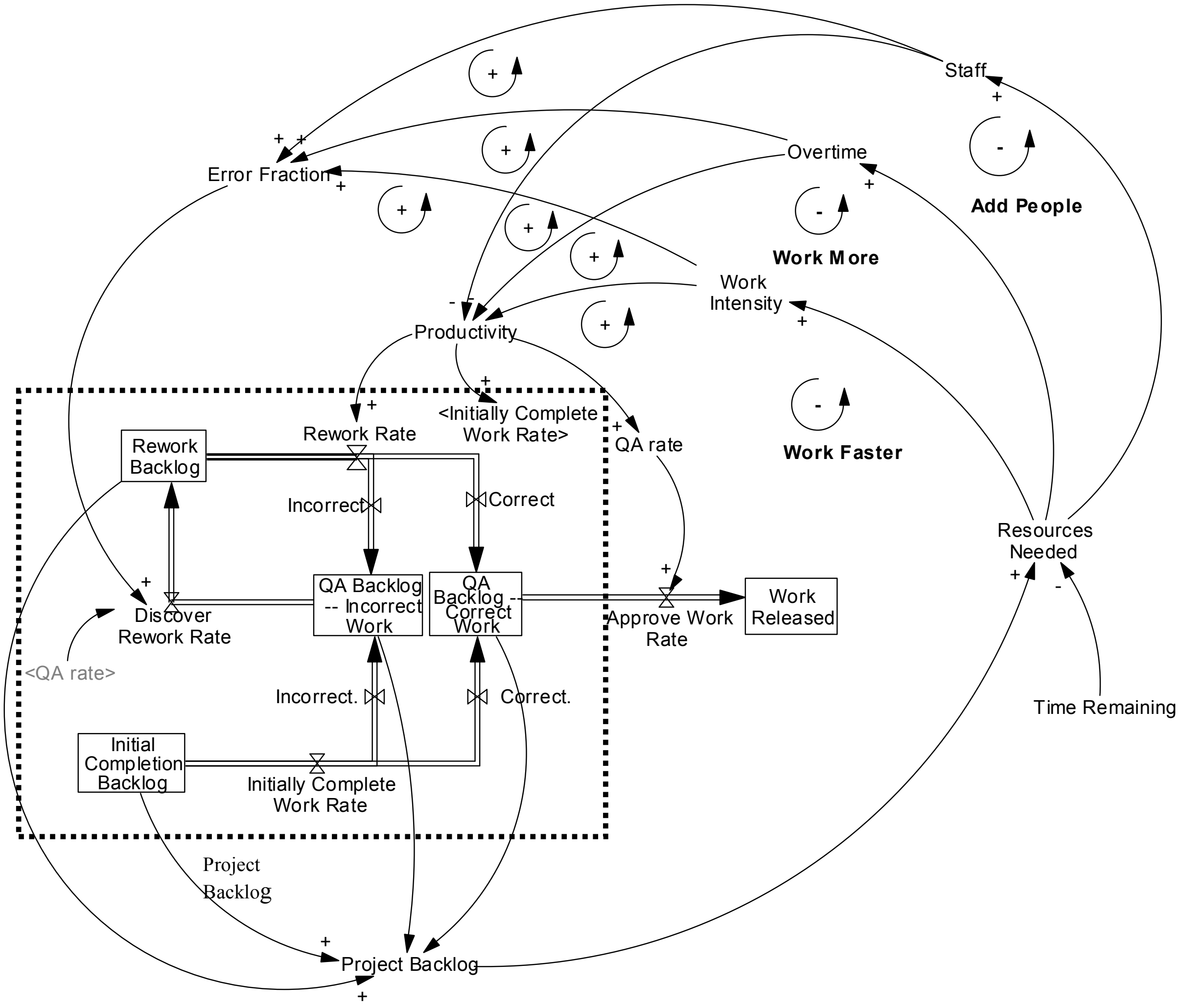

Figure 5: Policy resistance via ripple effects of rework and controlling feedback to improve schedule performance.

[7] The current work focuses on the three project control ripple effects on errors. Future work will investigate productivity ripple effects.

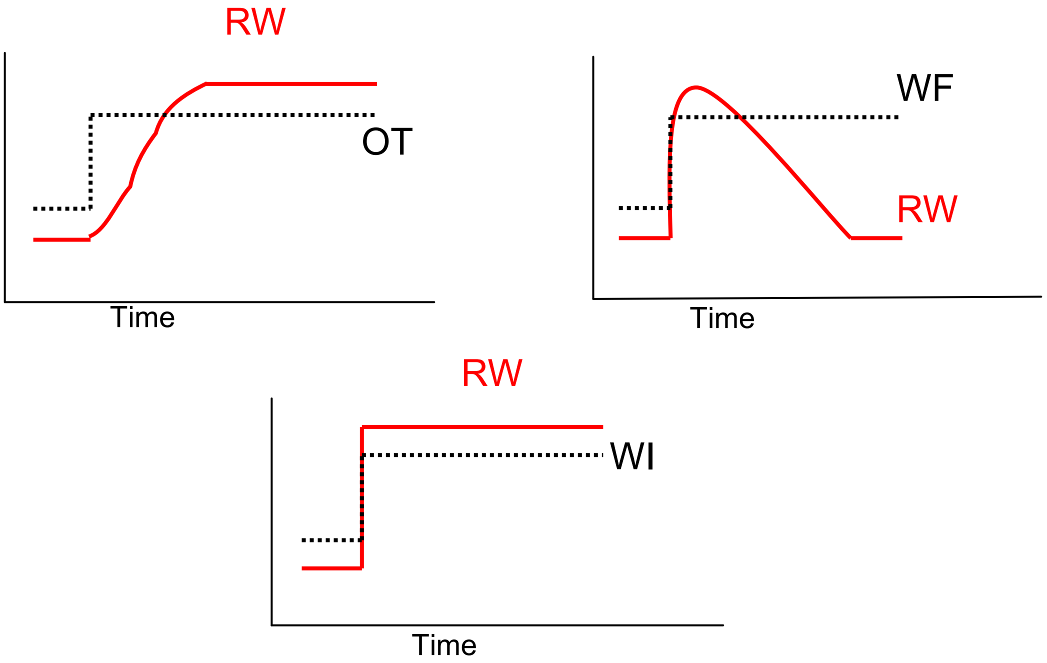

Figure 6: Impacts of overtime (OT), workforce size (WF), and work intensity (WI) on rework due to a step increase in the required effective workforce

Knock-on Effects. In addition to ripple effects, secondary (and tertiary) feedbacks emanate from the actions and reactions of project managers. Many feedbacks are a consequence 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 "human" reactions to project conditions such as purposefully hiding rework needs.

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 constraints allow. This can be caused by adding resources or exerting schedule pressure that can cause work to be done concurrently and/or out of the desired sequence, thereby reducing productivity and increasing errors. While there is planed concurrence on some projects, project problems can lead to the unplanned concurrence described here, with resultant adverse consequences (Cooper, 1994; Lyneis, et. al. 2001; Ford and Sterman 2003b).
- "Errors Build Errors" – undiscovered errors in upstream work products reduce the quality (i.e. contaminate) of downstream work as these unknown problems are built into downstream work products (Lyneis, et.al. 2001; Ford et al., 2004).
- "Errors Create More Work" – the process of correcting errors can increase the work required because fixing the errors adds tasks, or because fixing errors takes more effort than doing the original work. Ford and Taylor (2006) demonstrate that this feedback can create "tipping point" dynamics through which fraction complete can stop increasing and decline, often resulting in project cancellation.

- "Hopelessness" – Morale problems can exacerbate the effects of fatigue and rework, and create a sense of "hopelessness" that increases errors and reduces productivity. Hopelessness can also increase staff turnover.

How does this multitude of project structures impact project behavior? Although many behaviors are possible, Figure 7 summarizes the impact of the rework cycle and several of the most important of the feedback effects on project staffing. The rework cycle tends to delay project staffing, pushing the actual staff profile to the right, as errors are created, discovered, and reworked. This creates more errors which are discovered and reworked. Vicious circle feedbacks triggered by management responses to project problems tend to increase staffing, pushing project staffing (and effort applied through overtime) up, and to the right. This is due to productivity losses and increasing errors as a consequence of delays in recognizing and responding to project problems and the rework cycle.

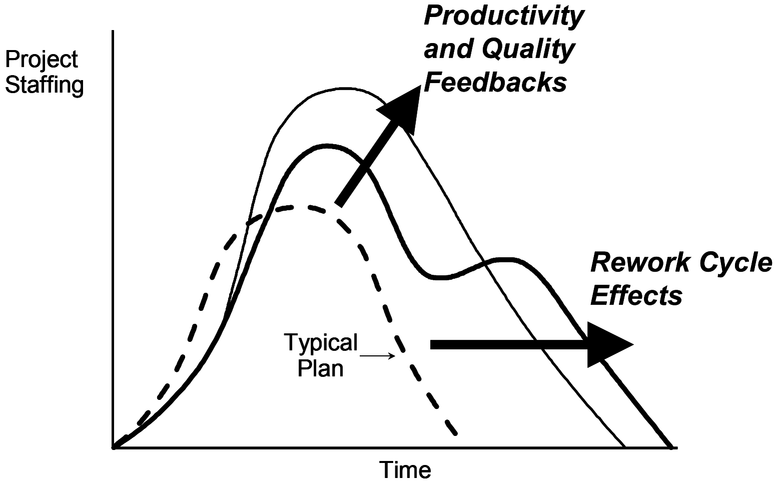

Figure 7: Rework cycle and productivity/quality effects on project staffing dynamics

How are the project dynamics created by the rework cycle, controlling feedback, ripple effects, and knock-on effects affected by various combinations of overtime, hiring and firing, and work pressure? How do different amounts, durations, and orders of their application impact project performance? More importantly, what combinations can bring a project back to meeting its targets with the least adverse consequences? What heuristics can improve project performance under a variety of different conditions? We address these questions in this research program. As a starting point, we have simplified the feedback structures in order to understand the impact of project controls on project dynamics. We will add additional feedback and relax assumptions in future work. The next section discusses the simplifications we have made in this first work iteration.

Representing the Project Dynamics Feedback Structure

For the first iteration of this research, we start with a simplified version of the rework cycle, negative feedback loops capturing all three project control actions, and one ripple effect feedback through error fraction for each management action. These structures are described in more detail below.

Project Workflow Sector

The workflow sector of the model simulates the completion of work required to complete the project. Work flow through the project is measured in units of "tasks." Conceptually a development task is an atomic unit of development work. Examples of development tasks include the selection of a pump, writing a line of computer code, and installing a steel beam. We assume tasks within a phase are uniform in size and are fungible. This assumption becomes more accurate as task size decreases. Therefore relatively small pieces of development work are selected as tasks. Fungible tasks are characteristic of some development project phases (e.g. the delivery and placement of soil for a roadbed). Many other development phases have interdependent but fungible tasks. Tasks are also assumed to be small enough to either require a change or be correct but not require a partial change.

The rework cycle used in this first analysis iteration is a slightly expanded version of that shown in Figure 2. Work flows among the stocks as shown in Figure 2 and as described above. The initial completion, quality assurance, and rework flows are constrained by either development processes or available resources. Development process constraints are based on the backlogs of available work and minimum task durations – this defines the "process rate" for each type of work.

The total amount of project control needed is measured by the gap between the amount of resources required to complete the remaining work by the deadline and the current resources. The one resource type (developers) are allocated to the three development activities proportional to the size of the backlog directly upstream of each activity. Additional assumptions in the work processes sector include:

- The same effort is required to perform initial completion, quality assurance, and rework (i.e. productivity for all the development activities are equal);
- Errors are all correctly identified in the QA process – no correct work is identified as containing errors, and no incorrect work slips by QA to be released; and
- The flow or work through the project is assumed to follow the critical path although no critical path is specified.

Project Controls Sector

We model three project resource controls that are commonly available to management: (1) exert pressure to increase progress (work intensity); (2) have developers work overtime (average workweek); and (3) hire additional staff. As illustrated in Figure 8, the actions are based on the effective workforce[8] deficit, which is the difference between the current effective workforce and the required effective workforce to complete the project by the deadline. The required effective workforce is the required work effort (in person-weeks) divided by the time remaining until the deadline[9]. Estimated work remaining is determined by the project scope less work released and

[8] The effective workforce is equal to the project workforce*overtime ratio*work intensity and is measured in "effective persons" or FTEs.

[9] When the computed time remaining falls below a threshold value, a constant "minimum time to complete the project" is used.

the portion of the QA work which is correct and will not need additional work.[10] The required effort is the work remaining (in tasks) divided by the resource productivity (in tasks/person-week).

The resource deficit, if positive, can drive project control action in one of the following ways. Either each of the three project controls can be used in isolation to fully respond to the resource deficit, or the deficit can be allocated between the three controls. The fraction of each control (overtime-only, hire/fire-only, and work intensity-only) to be applied is specified by a (currently exogenous) set of parameters that describe a project manager's control strategy. Multiplying the resource deficit by the fraction of the deficit to be filled by each control action allocates the required increase in effort to each of the three actions. Intuitively, a balanced, rational project control strategy has fractions that sum to 100%, under-reacting policies sum to less then 100%, and over-reacting policies sum to greater then 100%. After any delay, intensity, overtime, and people increase to fill their share of the deficit. Each project control is modeled as a first order exponential delay (Sterman 2000). For simplicity in this first analysis iteration, we assume that the delay in obtaining additional resources is the same for all three resource types.

[10] The actual estimation of work remaining is somewhat more complicated than shown in the figure as it also involves estimates of rework, the QA on that rework, and so on, and delays in perceiving progress. Further, early in the project budgeted scope and fraction of project time elapsed determine the estimate of work remaining because in practice it is often difficult to perceive the true state of the project. Nevertheless, the basic driver of estimated work remaining is work done compared to project scope.

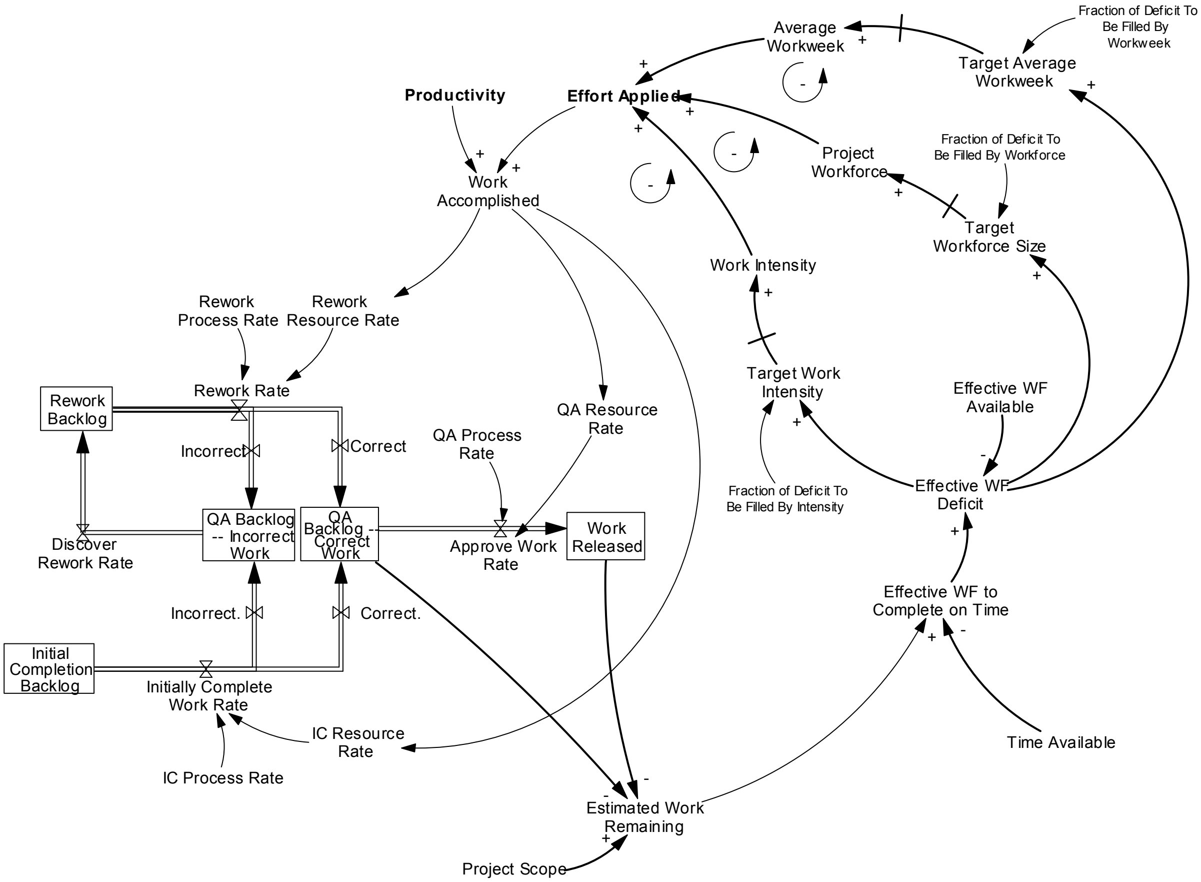

Figure 8: Management actions determined by work and time remaining (some variables and links not shown for clarity).

While the diagram in Figure 8 illustrates the complexity of the ripple and knock-on effects of management actions, in an effort to limit model complexity we have simplified the effects of project controls to impacts on the error fraction as shown in Figure 9. The effects of the three control actions operate in slightly different ways:

- Work Intensity – An increase in work intensity is assumed to inadvertently cause developers to create more errors. Therefore incremental errors (errors in addition to the base project error fraction) from work intensity are driven directly, without delay, by the amount by which work intensity exceeds normal intensity (Figure 6).
- Overtime – Overtime is assumed to increase errors due to fatigue, which occurs with sustained overtime. Therefore, overtime is delayed to model fatigue (Figure 6). Incremental errors from overtime are proportional to the “sustained” overtime over normal overtime. So, except for the delayed response, this formulation is identical to the work intensity formulation.
- Workforce – Added staff are assumed to have less knowledge and experience than the existing staff. The incremental errors from increased workforce depend on the number of unoriented workers and the additional errors they produce. Incremental errors are based on the relative errors created by unoriented workforce and their share of the total

workforce. As shown in Figure 6 once unoriented workers become oriented the error fraction returns to its base value.

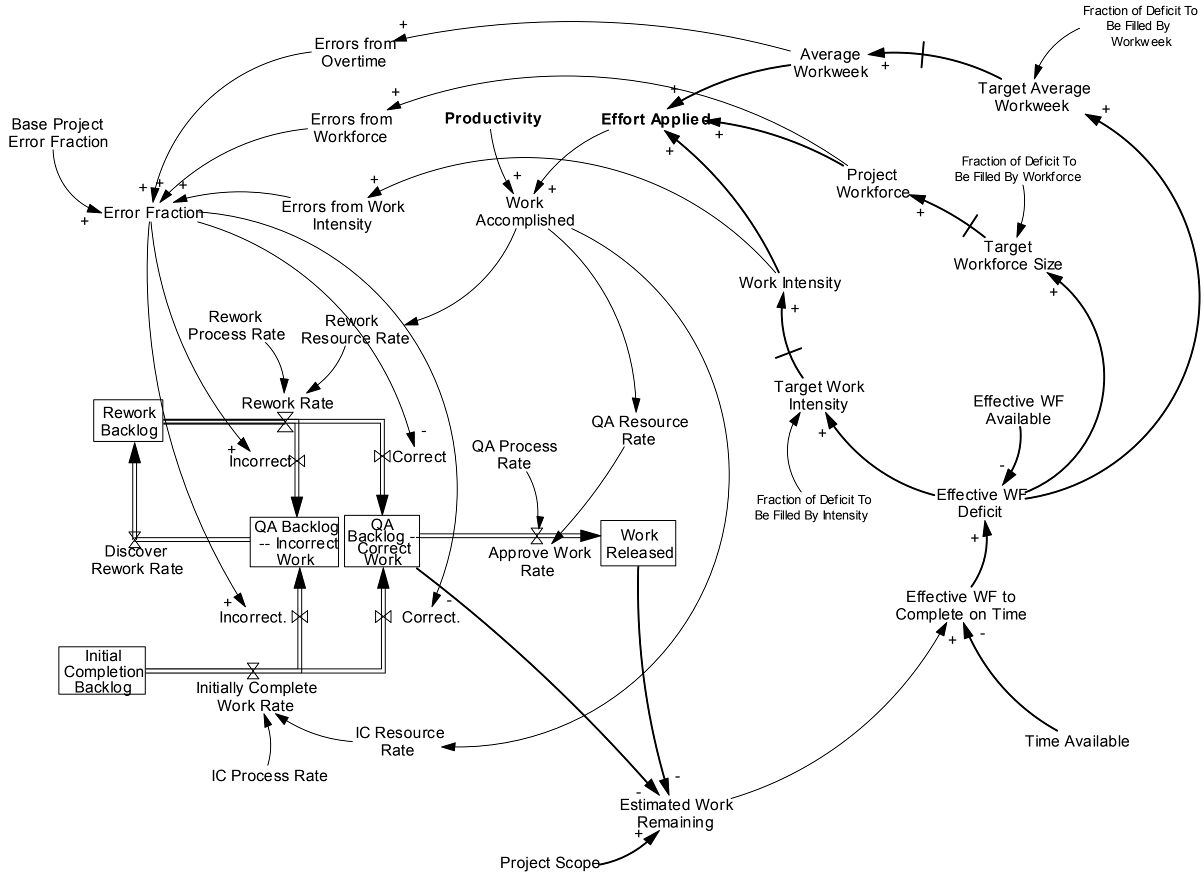

Figure 9: Management action create ripple effects through error fraction

Additional assumptions of the project controls model sector include:

- “Expert project manager” assumption (manager immediately recognizes changes in rework, scope changes, project progress, accurately forecasting resource needs, accurately measuring error fractions, etc.);
- Project controls cannot be reduced below their initial values; and
- Project deadline is exogenous and fixed.

Model Validation and Testing

The workflow structure of the model was validated using standard methods for system dynamics models (Sterman, 2000). Basing the model on previously validated 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 zero and simulating project behavior. As expected, no work was performed. The results of other extreme conditions testing are consistent.

The model’s behavior for typical conditions is consistent with previous project models and practice (e.g. the "S" shaped growth percent complete shown in Figure 10). Model behavior was also compared to actual project behavior as described by Ford (1995), Ford and Sterman (1998 and 2003b), Lyneis et al. (2001), Lee et al. (2005), and Taylor and Ford (2006) and found to closely match the behavior modes of actual projects. Based on these tests the workflow model sector was assessed to be useful for the project control investigation.

The project controls portion of the model was also tested using standard methods for system dynamics models as described in Sterman (2000). The model is dimensionally consistent and was subject to extreme conditions testing (e.g. when the overtime control is turned off overtime is not implemented to correct a troubled project). Many of the components are standard modeling techniques as described in Sterman (2000). For example, the change in resources applied by each the project control is adjusted using a first order exponential delay. The behavior of the model is consistent with expected system behavior. For example, when a resource deficit becomes apparent the model uses the enabled project control to adjust resources to address the deficit and then release resources once the deficit is addressed.

The reference case project used in this investigation has 500 tasks to be completed by 20 people. The reference rework fraction for the project is 20%. The project has a completion deadline of week 65 and a budget of $1.29 million[11]. The evolution of project percent complete is shown in Figure 10. Without utilizing any of the three project controls the project is completed at week 65 for a cost of $1.29 million. Therefore, the base case project requires rework but has just enough time and money to complete the project on time and in budget without any project controls.

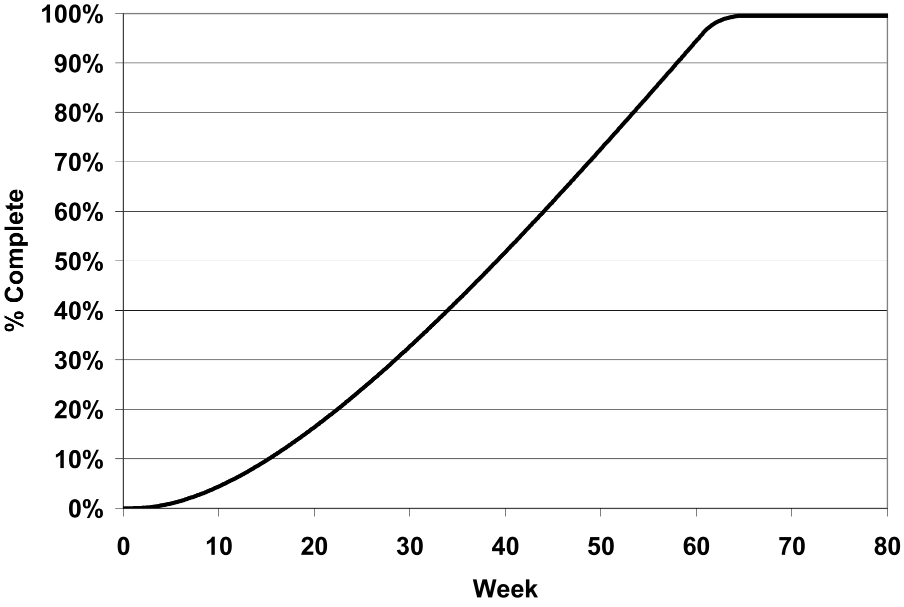

Figure 10: Base case project percent complete

[11] Project costs are calculated based solely on resource cost. The standard rate per resource (i.e. developer) is $1,000 per week. Overtime hours are paid at a time and a half rate. There is no additional cost associated with work intensity.

The reference case project was then simulated with each of the project controls activated. In all three cases the project was completed at week 65 with an identical evolution of percent complete shown in Figure 10 and at a cost of $1.29 million. This identical performance to the base project was expected because the base case project was able to meet its required completion date and therefore did not employ any of the three enabled project controls.

Model Use and Error Impact

To investigate the effectiveness of the three project controls in correcting a poorly performing project the reference case project (Figure 10) was subjected to a scope change equal to 20% of the original project scope at week 30. [12] The scope change causes the project to be completed at week 77, missing the 65 week deadline by 18% (Figure 11).

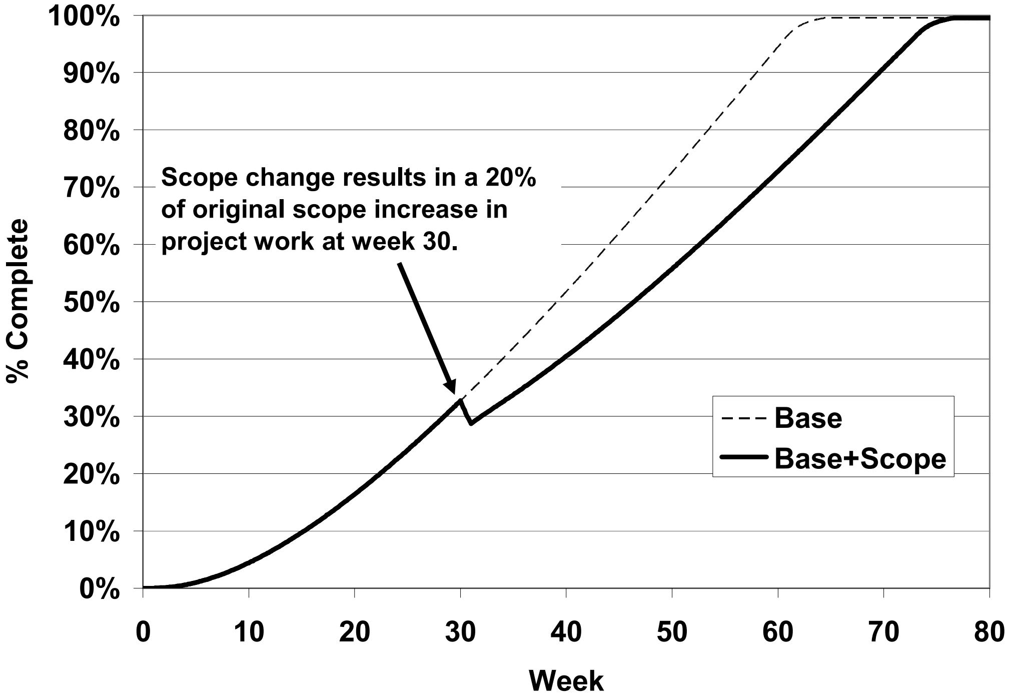

Figure 11: Percent complete for project experiencing scope change at week 30

Each of the three project control policies was then used to attempt to complete the project by the required completion date. For this analysis the three project controls are assumed to have equal application delays times (1/8 of a week) and no impact on project rework. The performance of the project controls is summarized in Table 1. The evolution of percent complete for the project with each of the three project controls enabled individually, the three project controls assigned simultaneously with each assigned 33% of the effective workforce deficit, and the project with no project control enabled is shown in Figure 12. The effective workforce for the four simulations utilizing project controls to address the scope change is shown in Figure 13.

[12] As used here percent complete is the work release/project scope. When the project scope is increased by 20% the percent complete only drops approximately 5% because this scope increase immediately changes the denominator in the percent complete fraction but the numerator is not immediately impacted.

Table 1: Results summary for project subjected to scope change and project management[13]

Project Simulation	Duration (weeks)	Cost ($1,000,000)	% Schedule Improvement from Ref Project (No PC)	% Cost Increase over Ref Project (No PC)
Ref	65	1.29		
Ref + Scope Change (No PC)	77	1.53		
Ref + Scope Change + OT	67	1.63	13.0%	+6.5%
Ref + Scope Change + WF	67	1.53	13.0%	0.0%
Ref + Scope Change + WI	67	1.34	13.0%	-12.4%
Ref + Scope Change + 33%	67	1.50	13.0%	-2.0%

Note: "Ref" = Reference

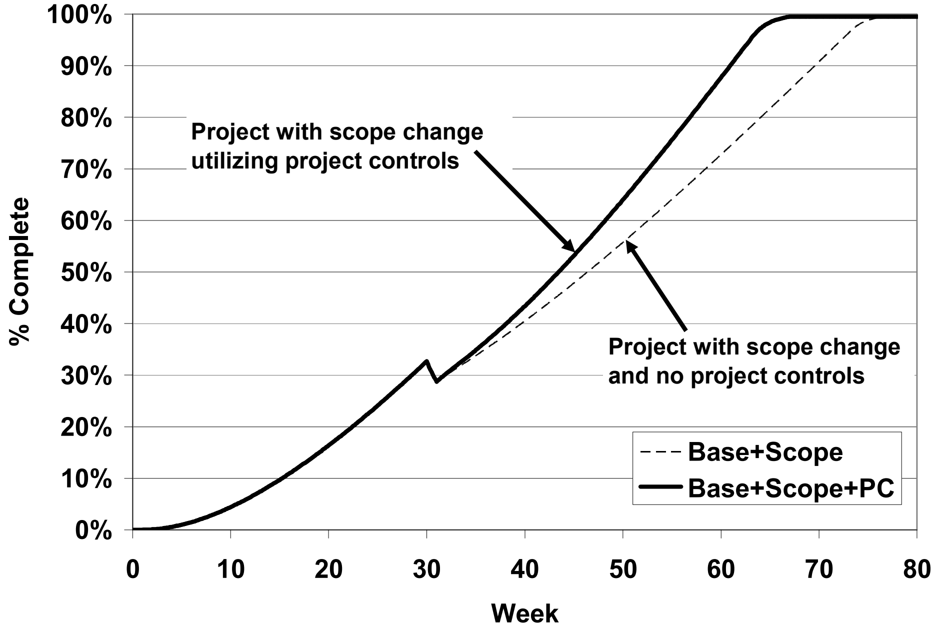

Figure 12: Percent complete for two projects subject to scope change[14]

[13] "PC" = project controls, "OT" = overtime, "WF" = workforce, "WI" = work intensity, and "33%" = effective workforce deficit addressed equally by all three project controls.

[14] The evolution of percent complete for the three projects utilizing each of the project controls is identical.

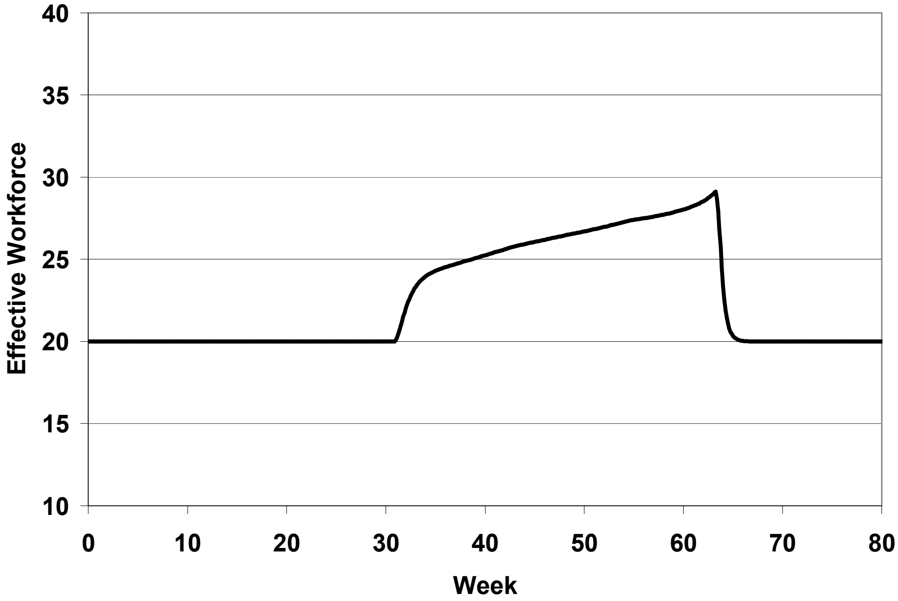

Figure 13: Effective workforce applied used to correct project experiencing scope change

Table 1 and Figure 12 reveal that the use of project controls to overcome the scope change during the course of the project result in all three projects being completed at week 67 and 13% faster then the no project control project. Since the response delay times for the three project controls are identical, the response to the scope increase from the three project controls is identical, as evidenced by the identical behavior modes for the change in effective workforce for project controls shown in Figure 13. This identical response is also due to the assumption that there are no additional errors due to the use of the three project controls. The impact of errors due to project controls is investigated next.

Impact of Errors from Project Controls

But no project can actually garner the benefits described above. This is because, as discussed earlier, the use of project controls have negative impacts on project rework. Fatigue due to excessive overtime, pressure to work faster, and the incorporation of inexperienced workers can increase the amount of rework on a project. To examine the effect of additional errors introduced by the use of the project controls we simulate the reference case project that experiences scope change and again use the three project controls to manage the project. However, we allow the project controls to introduce additional errors when utilized[15]. The evolution of percent complete for this analysis is shown in Figure 14. A summary of the schedule performance is shown in Table 2.

[15] Additional errors are enabled by setting the value of the “fraction of additional work due to OT that requires rework” = 1, the “fraction of correct work fraction lost due to unoriented WF” = 1, and the “fraction of additional work due to WI that requires rework” = 1 in the model.

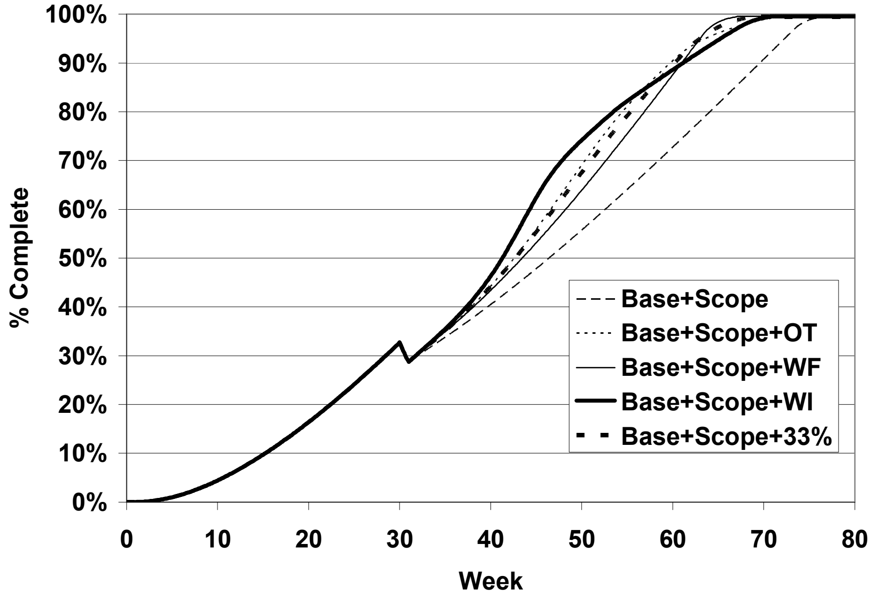

Figure 14: Percent complete for base project with scope change for project controls with additional errors

Table 2: Results summary for project subjected to scope change and project management with errors

Project Simulation	Duration (weeks)	Cost ($1,000,000)	% Schedule Improvement from Ref Project (No PC)	% Cost Increase over Ref Project (No PC)
Ref	65	1.29		
Ref + Scope Change (No PC)	77	1.53		
Ref + Scope Change + OT + Errors	72	2.35	6.5%	+53.6%
Ref + Scope Change + WF + Errors	67	1.54	13.0%	+0.7%
Ref + Scope Change + WI + Errors	71	1.42	7.8%	-7.2%%
Ref + Scope Change + 33% + Errors	69	1.73	10.3%	+13.1%

As shown in Figure 14 and Table 2, the introduction of additional errors by project controls has a noticeable impact on project schedule performance. Without additional errors the project controls are able to complete the project experiencing scope change within two weeks of the deadline (Table 1). When project controls introduce additional errors the project using the overtime policy is completed seven weeks past the deadline, the project using the workforce policy is completed two weeks past the deadline (the same week as the projects using project controls with no errors, Table 1), the project utilizing work intensity is completed six weeks past the deadline, and the project using all three project controls is completed four weeks past the deadline. The overtime project had the worst cost performance followed by the project using all three project controls, the project utilizing workforce, and the best cost performer, and the only project that came in below the cost of the no project control project is the project using work intensity. For practitioners, the overtime, workforce, intensity, and balanced policies improve schedule performance over no controls (with error impacts) by 6.5%, 13.0%, 7.8%, and 10.3%, respectively (Table 2)

A shift in feedback loop dominance from the negative project control loops to the positive rework-generating loops cause the slowdown in progress late in the project. The variation in performance of the three project controls in overcoming the scope change is related to the impact the three controls have on the project rework fraction. For all three project controls, the effective workforce applied increases after the scope change in initiated in week 30 (Figure 15). As the project approaches the deadline the effective workforce increases dramatically with the work intensity, resulting in a larger effective workforce then the overtime, workforce, and "33%" policies. The difference in the effective workforce for the three project controls is partially due to their different impacts on the rework fraction. The change in rework fraction of each of the three project controls in shown in Figure 16 (the base rework fraction is 0.2).

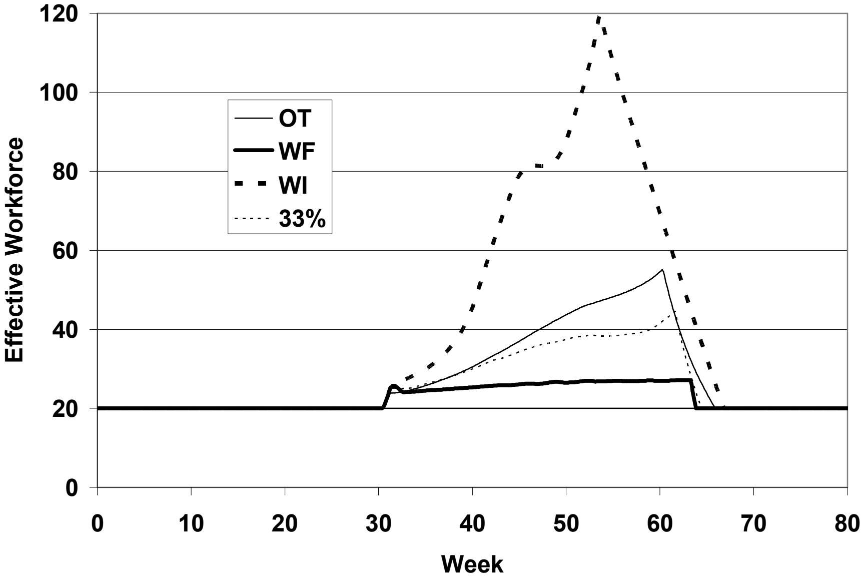

Figure 15: Change in effective workforce for four project controls

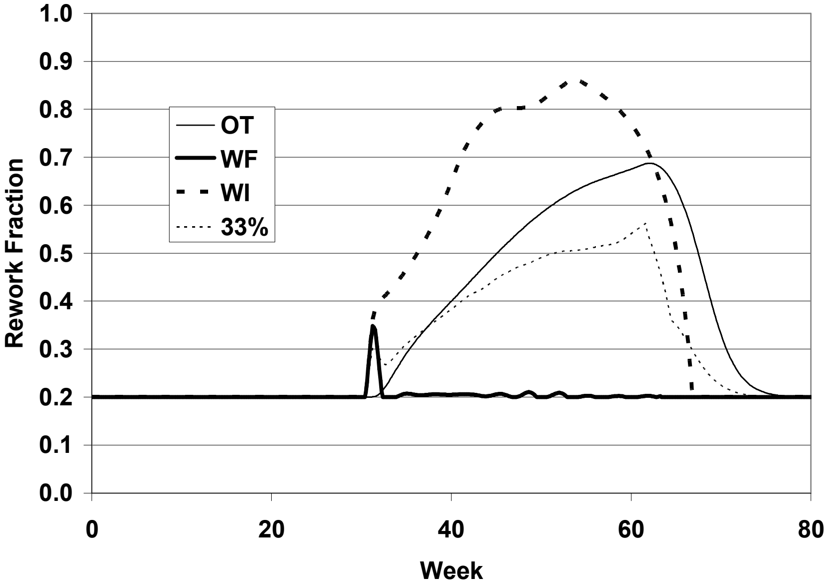

Figure 16: Change in rework fraction for four project controls

As shown in Figure 16 increasing the workforce creates an immediate increase in errors from the new employees. As employees gain experience the rework fraction returns to the nominal value relatively quickly when compared with the other projects. Work intensity causes an immediate and growing increase in the rework fraction as work intensity builds. Overtime yields a delayed increase in the rework fraction as there is a delay for fatigue to begin to affect the workforce. The changes in rework fraction in response to the three project controls create different quantities of total rework on the project. The quantity of rework for each project control, the 33% project, and the reference case project are shown in Figure 17. Figure 17 reveals that the use of work intensity produces the greatest quantity of total rework followed by overtime and finally increasing the size of the workforce. For all four projects utilizing project controls the use of project controls produces more tasks requiring rework then the project that utilizes no project controls.

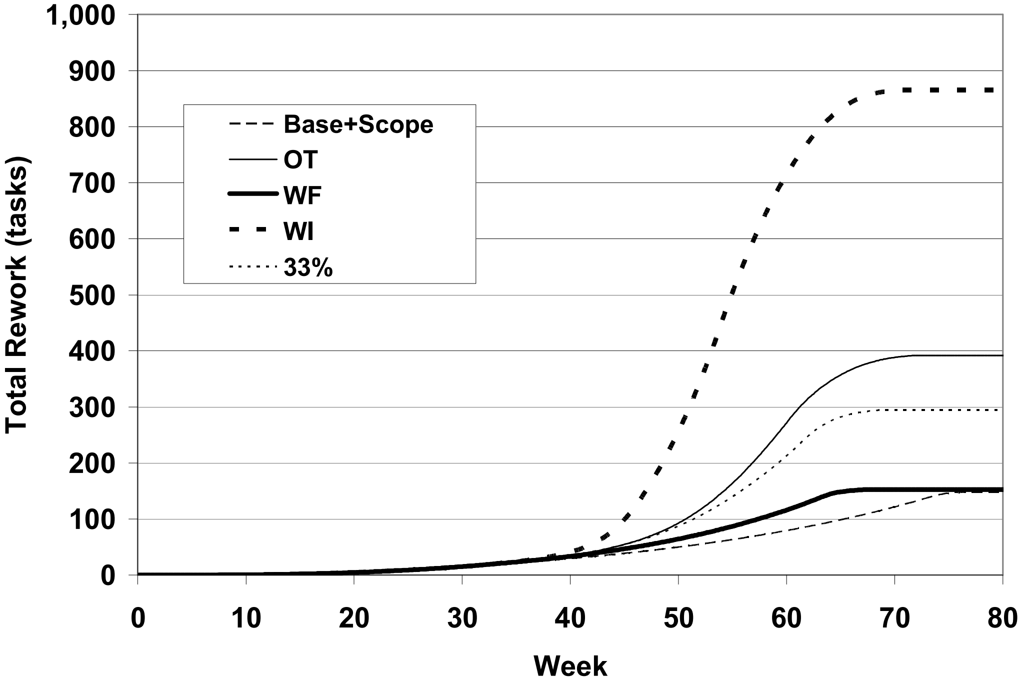

Figure 17: Total tasks requiring rework for each project control

Impact of realistic project control delays on project performance

The previous section described the impact of additional errors introduced by project controls but did not include the impacts of realistic delays in the application of the three project controls. Generally work intensity can be increased rather quickly where as adding overtime and hiring new workers can require more time to implement. To simulate more realistic delays in the application of project controls the application delays for the three project controls where set as follows; OT = 2 weeks, WF = 6 week, and WI = 1 day. The evolution of percent complete for this analysis is shown in Figure 18. A summary of the schedule and budget performance is shown in Table 3.

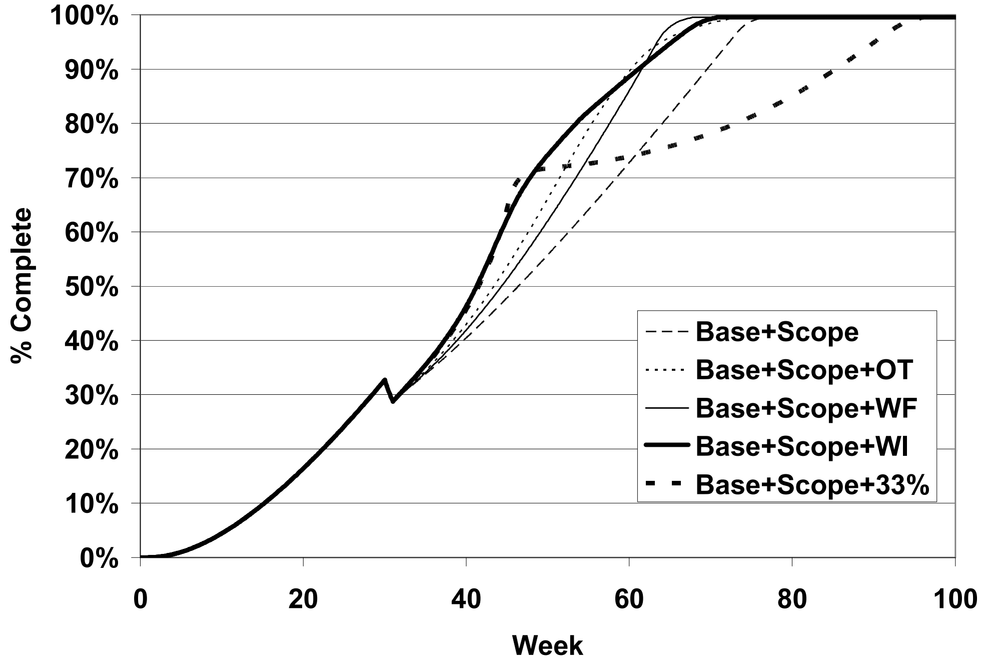

Figure 18: Percent complete for base project with scope change, additional errors, and realistic project control delays

Table 3: Results summary for project subjected to scope change, project controls, side effects, and realistic project control delays

Project Simulation	Duration (weeks)	Cost ($1,000,000)	% Schedule Improvement from Ref Project (No PC)	% Cost Increase over Ref Project (No PC)
Ref (No PC)	65	1.29		
Ref + Scope Change (No PC)	77	1.53		
Ref + Scope Change + OT + Errors + Delays	74	2.36	+3.9%	+54.2%
Ref + Scope Change + WF + Errors + Delays	68	1.64	+11.7%	+7.1%%
Ref + Scope Change + WI + Errors + Delays	71	1.42	+7.8%	-7.2%%
Ref + Scope Change + 33% + Errors + Delays	96	2.13	-24.7%	+39.2%

As shown in Figure 18 and Table 3 the addition of more realistic delays has no impact on work intensity (the adjustment delay remained 1 day) but degrades the performance of overtime and work force adjustments. The more interesting case is the performance of the project that employees all three project controls equally (the “33%” case). Counterintuitively, the balanced policy performs worst. Initially this project progresses quickly after the scope change. However, the delays associated with the side effects from the three project controls combine to push the rework fraction close to 1 (Figure 19) causing the sudden slowing of project progress show between weeks 50 and 60 in Figure 18. The timing of the project control side effects cause the project to experience more then twice the amount of rework (Figure 20) as the project experiencing equal project control application delays (Figure 17).

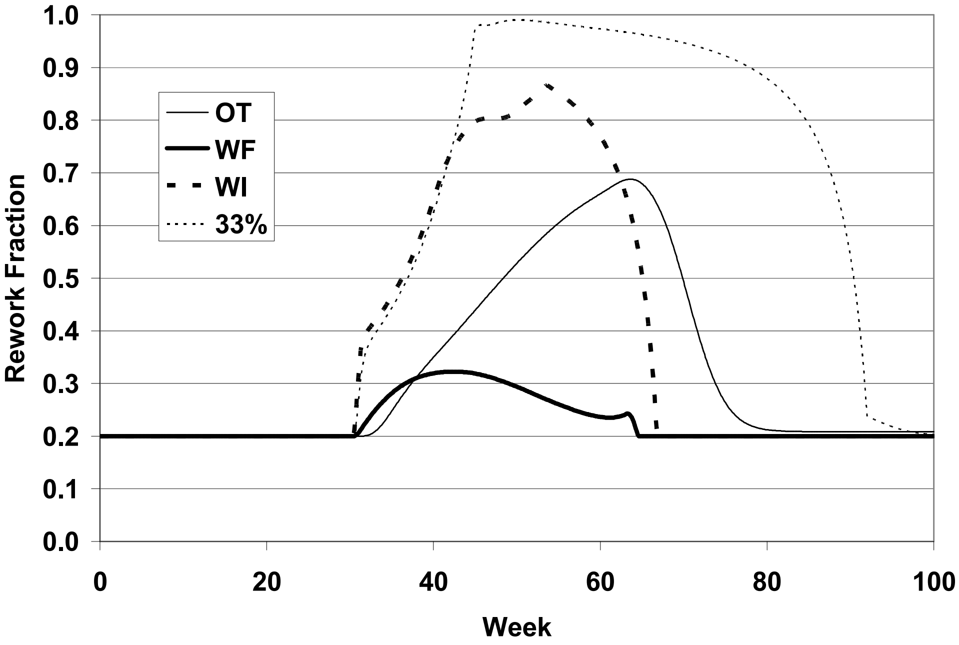

Figure 19: Change in rework fraction for four projects

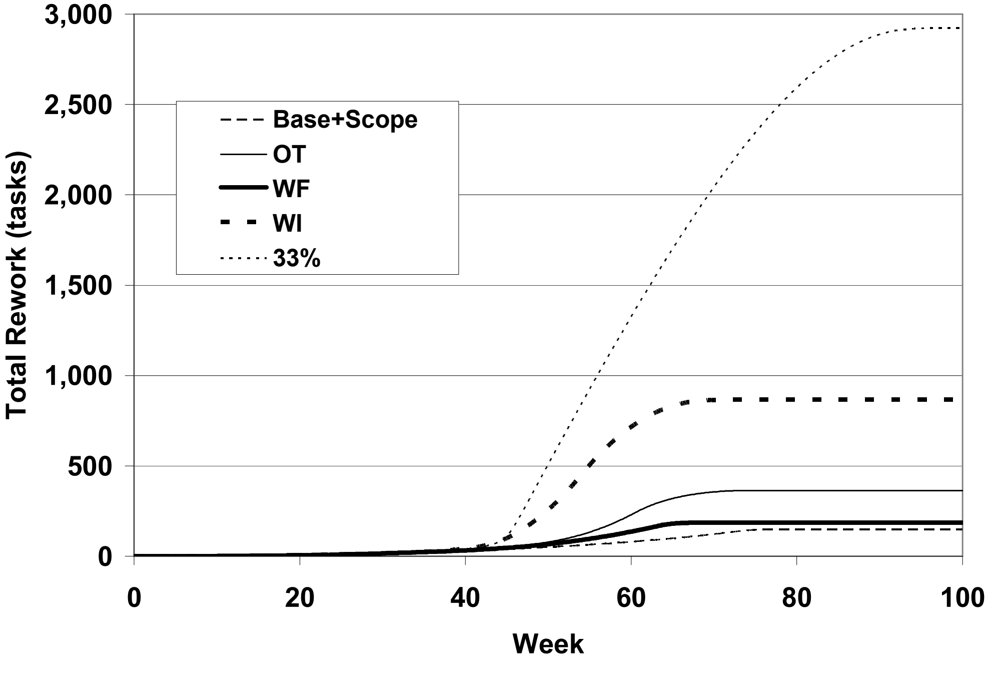

Figure 20: Total rework for four projects utilizing project controls and realistic delays

Figure 17 and Figure 20 display an interesting paradox for project managers. Despite the increase in the effective workforce for the projects that utilize project controls, the actual work performed (project scope + total rework) is greater then the base case project with scope change that utilized no project controls. The projects utilizing project controls is finished faster than the reference case project but require the completion of a greater amount of rework to complete.

Sensitivity Analysis of Error Impact

To investigate the impact of errors introduced by project controls a single variant sensitivity analysis to additional errors introduced by the three project controls was performed. The analysis varied parameters within the model structure that regulate the amount of errors introduced by each project control. The sensitivity analysis was performed utilizing the base case project + scope change (Figure 11).

To investigate the impact of additional errors introduced by overtime the project's fraction of additional work requiring rework due to overtime was varied from 0 to 2. This model parameter regulates how an increase in the overtime ratio increases the rework fraction. The results of the sensitivity analysis are shown in Figure 21.

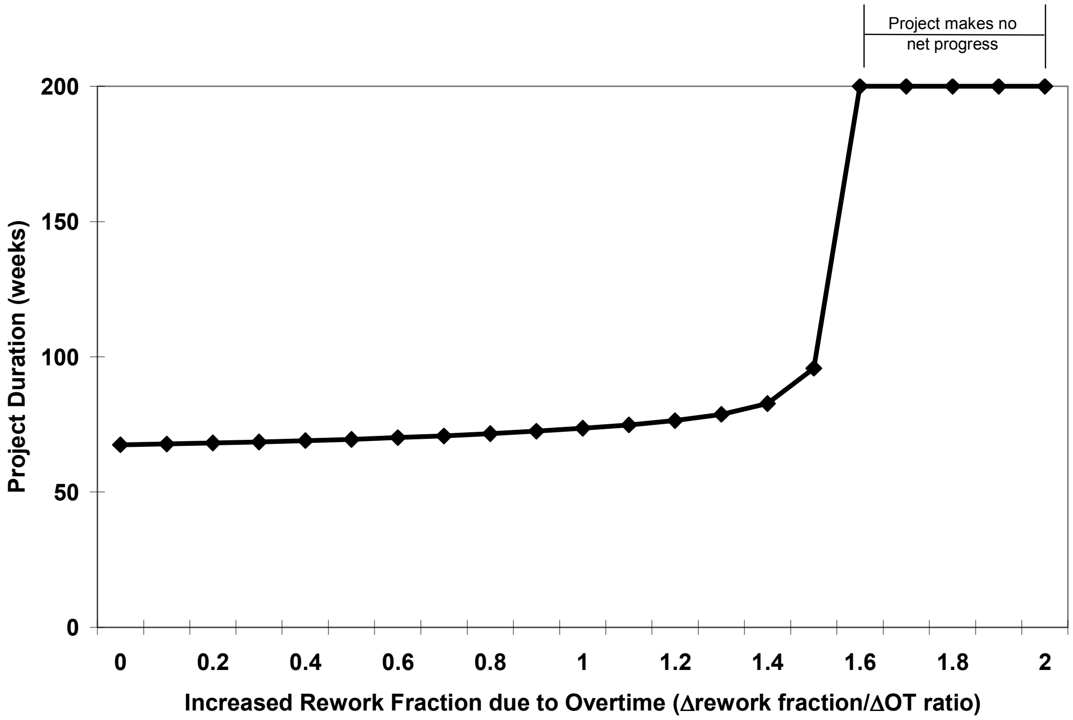

Figure 21: Sensitivity analysis of errors introduced by overtime

The shape of Figure 21 reveals an exponential relationship between project duration and the sensitivity of the project to errors introduced by overtime. Lower sensitivity levels of rework to overtime have a minimal impact on project duration represented by the relatively flat line on the left side of Figure 21. As sensitivity to overtime increases, the impact of additional errors due to overtime on project duration increases dramatically as shown by the rapidly increasing slope of the line in the middle portion of Figure 21. At a fraction of 1.6 the errors introduced by fatigue due to overtime push the rework fraction to 100%, the project is overwhelmed by rework, and the project duration increases indefinitely[16]. At 100% rework the project is circulating all work

[16] The graph in Figure 21 "flat-lines" at a duration of 200 weeks because simulation time is 200 weeks.

through the rework cycle and is not releasing any work, therefore it cannot be completed regardless of the time available to finish the project.

To investigate the impact of errors introduced by increased work intensity the project's fraction of work requiring rework due to overtime is varied from 0 to 2. The results of the sensitivity analysis are shown in Figure 22.

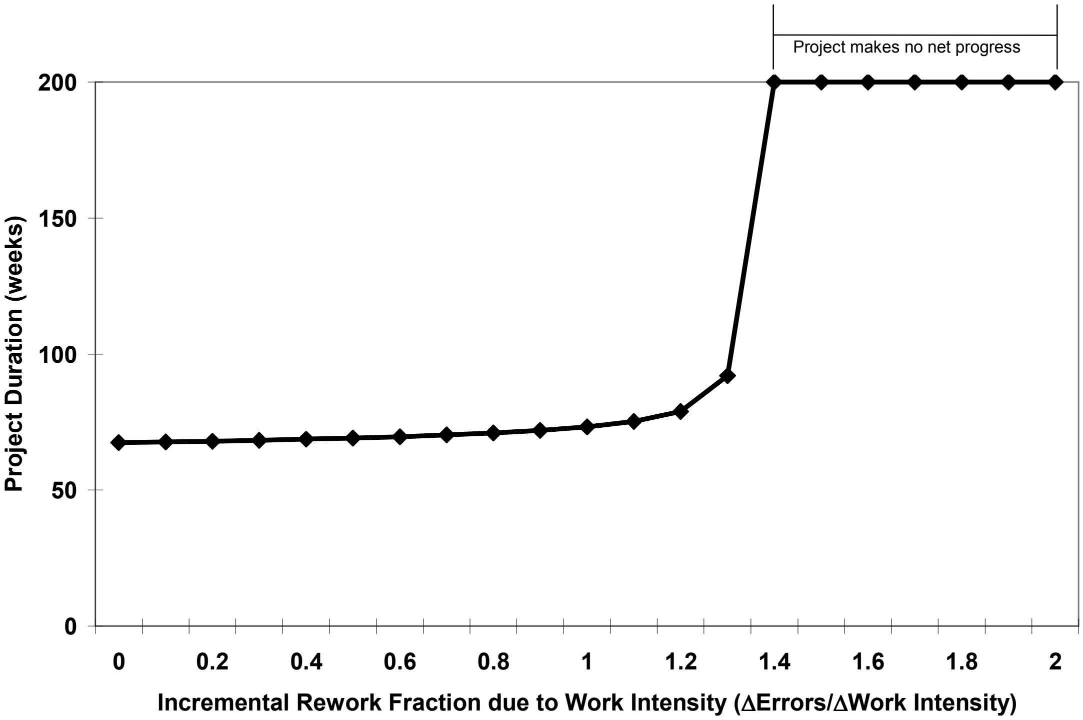

Figure 22: Sensitivity analysis of errors introduced by overtime

The shape of Figure 22 reveals an exponential relationship between project duration and the incremental errors introduced by increased work intensity. Small fractions have a minimal impact on project duration represented by the relatively flat line on the left side of Figure 22. As the incremental errors introduced by work intensity increases, the impact of these errors on project duration increases dramatically as shown by the rapidly increasing slope of the line in the middle portion of Figure 22. Once the fraction reaches 1.4 the errors introduced by work intensity push the rework fraction to 100%, the project is overwhelmed by rework, and the project duration increases indefinitely[17]. At 100% rework the project is circulating all work through the rework cycle and is not releasing work, therefore it cannot be completed regardless of the time available to finish the project.

To investigate the impact of errors introduced by the addition of inexperienced staff to the project the relative error fraction of the unoriented workforce is varied between 0 and 2. The results of the sensitivity analysis are shown in Figure 23.

[17] The graph in Figure 22 "flat-lines" at a duration of 200 weeks because simulation time is 200 weeks.

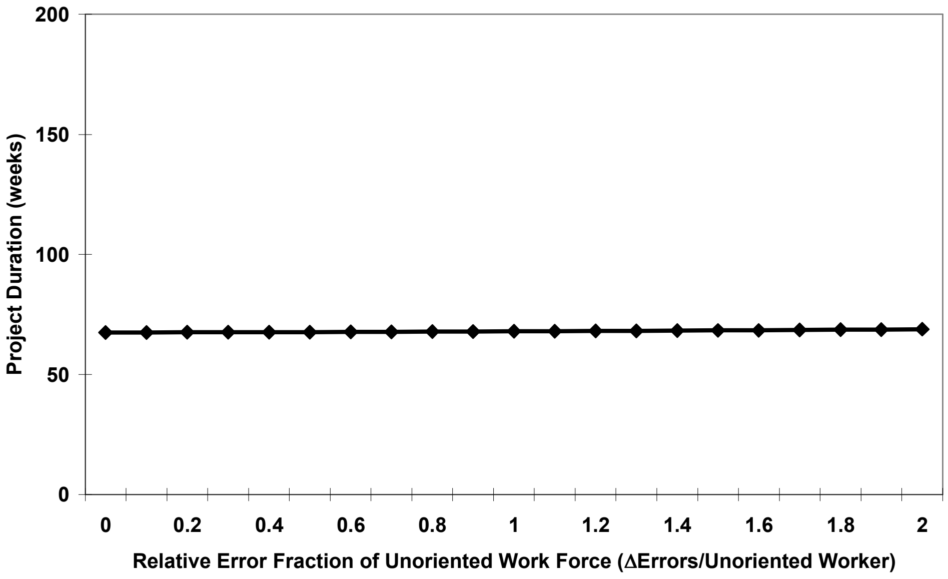

Figure 23: Sensitivity analysis of errors introduced by unoriented workers

The flat slope of the line in Figure 23 reveals that project duration is relatively insensitive to the errors introduced by adding additional workers to the project[18]. Unlike the impact of errors introduced by overtime and work intensity, the errors introduced by adding additional workers to the project does not cause the rework fraction to reach a perpetual value of 1, trapping the project in the rework cycle.

The relative insensitivity of project duration to errors introduced by unoriented workers as opposed to the errors introduced by overtime or increased work intensity is related to the evolution of errors due to the three project controls (Figure 6). The sustained use of overtime and work intensity to correct project schedule performance results in a sustained increase in errors due to the use of the two project controls. As currently modeled, the developer does not develop a resiliency to the negative effects of overtime or work intensity after working a set amount of time. As long as the developer continues to work hours beyond the normal work week or under increased pressure, the developer will continue to make addition errors due to the stress caused by these two project controls. Time itself does not reduce the errors introduced from the use of these two project controls. In contrast, the additional errors introduced by increasing the size of the project workforce does decrease with time. As time progresses inexperienced developers become more proficient and they non longer produce errors due to their inexperience.

Conclusions and Further Research

The current work presents initial work on examining the impact of three project controls (overtime, work intensity, and increased workforce) on project budget and schedule

[18] This insight has been confirmed by sensitivity testing of larger values (up to 100) of incremental errors due to unoriented workforce.

performance. The work develops and demonstrates a system dynamics project model that simulates the impact of three labor resource project controls on schedule performance.

This work has several implications for researchers. It presents a new model structure that simulates the impact of three project controls on rework and project schedule performance. This structure could be applied to other project models to investigate the impact of the three project controls on other issues such as work quality. The work also begins to fill the gap of providing policy advice to managers based on the results of project modeling. These policies can be further tested in future research for effectiveness in improving other performance measures such as project quality.

This work also has implications for managers. The heuristics developed in extensions of the planned work will provide managers with guidance on how and when to implement the three project controls modeled. This will allow managers to maximize project performance when the three project controls are utilized. The work will also provide managers insight into the expected project behavior when project controls are implemented. For example, the increased rework associated with the hiring of additional workers will be temporary as opposed to the sustained rework impacts of overtime and increased work intensity.

The model presented in the current work is at the initial stage of development. The project controls model sector will be further validated. The detrimental impacts of the three project controls on productivity will be investigated in a similar manner as the rework investigation presented here. The model will be expanded to allow the use of project controls to be driven by budget performance in addition to schedule performance. The improvement and expansion of the model and the test results will allow us to develop heuristics for the use of the three project controls. For example, perhaps managers should employ work intensity and overtime to overcome short-term problems and utilize additional workers for long-term problems due to the sustained effects of overtime and work intensity on rework.

The work presented here is a step towards developing an improved understanding of how managers impact project performance through their use of project controls. The model presented in this work can be expanded and improved to investigate additional issues. The impact on project performance of including non-linear functions into the relationships between the three project controls and errors and productivity can be investigated. The impact of the three project controls on projects subject to tipping point dynamics as described by Taylor and Ford (2006) can also be investigated. This investigation could investigate the effectiveness of the three project controls in "saving" a project that has crossed the tipping point.

References

Cooper, K.G. 1993. "The rework cycle: why projects are mismanaged." *PM network*. February: 5-7.

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

Evans, M. 2005. “Overdue and over budget, over and over again; project management.” *The Economist*, 375 (8430): 66.

Flyvbjerg, B., Skamris Holm, M, and Buhl, S. 2003. “How common and how large are cost overruns in transport infrastructure project?” *Transport Reviews*, 23(1): 71-88.

Ford, D. 1995. The Dynamics of Project Management: An Investigation of the Impacts of Project Process and Coordination on Performance. doctoral dissertation. Massachusetts Institute of Technology, Cambridge, MA. 1995.

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

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

Ford, D., Lander, D., and Voyer, J. 2004. “Business strategy and real options in the context of large engineering projects.” *Journal of Global Competitiveness*. 12(1): 1-9.

Lee, S., Pena-Mora, F., and Park, M. 2005. “Quality and change management model for large scale concurrent design and construction projects.” *ASCE Journal of Construction Engineering and Management*. 131(8): 890-902.

Love, E., Mandal, P., Smith, J., and Li, H. 2000. “Modelling the dynamics of design error induced rework in construction.” *Construction Management and Economics*. 18: 567-574.

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.

Lyneis, J. and Ford, D. 2007, “System dynamics applied to project management: A survey and directions for future research.” *System Dynamics Review*. 23(4).

Matta, F. and Ashkenas, R. 2003. “Why good projects fail anyway.” *Harvard Business Review*, 81(9): 109 – 114.

Nassar, K., Nassar, W., and Hegab, Y. 2005. “Evaluating cost overruns of asphalt paving projects using statistical process control methods.” *ASCE Journal of Construction Engineering and Management*, 131(11): 1173-1178.

Reichelt, L. and Lyneis, J. 1999. “The dynamics of project performance: Benchmarking the drivers of cost and schedule overrun.” *European Management Journal*. 17(2): 135-150.

Sterman, J.S. 2000. *Business Dynamics: Systems Thinking and Modeling for a Complex World*. Irwin McGraw-Hill, New York.

Taylor, T. and Ford, D. 2006. “Tipping point failure and robustness in single development projects.” *System Dynamics Review*. 22(1): 51-71.