Building on Shifting Sands: The Structure of Repetitive IT Project Escalation, Crisis, and De-escalation

Andrew Whitmore
University at Albany
State University of New York
College of Computing and Informatics
1400 Washington Avenue
Albany, NY
Aw955955@albany.edu

Eliot Rich
University at Albany
State University of New York
School of Business
1400 Washington Avenue
Albany, NY
rich@acm.org

Mark R. Nelson
Digital Content Strategist
National Association of College Stores
500 East Lorain Street
Oberlin, OH 44074
mnelson@nacs.org

Abstract

This research examines a case of extended failure to complete a critical and complex IT modernization effort in a US government organization. The project has been revamped, stopped and restarted several times, and as of the writing of this paper has not completed. From the system dynamics perspective, the problem appears to be related to a dynamic and repeating decision and management process with an embedded project management model. We hypothesize that the cyclical project escalation and abandonment is due to the continuous introduction of new requirements during the project lifecycle. A simulation model is developed to test the hypothesis and the results support the proposition that scope creep resulting from the introduction of new requirements may be a causal factor in the cycles of project escalation and de-escalation. The model is then used to test a series of policy options that are aimed at mitigating these cycles. Conclusions, recommendations, and limitations are discussed.

Keywords: Project Management, Government Information Systems, Project Commitment

Introduction

This paper examines the phenomenon of cyclical IT project escalation and abandonment at the U.S. Internal Revenue Service. From the system dynamics perspective, the problem appears to be related to a dynamic and repeating decision and management process with an embedded project management model. We decided to use well understood project models to develop this perspective. It has been advocated that simple models may be used as a mechanism to introduce the concepts of stock and flow to unfamiliar audiences (Richardson and Andersen 1995) and establish grounds for future communications. Here we present our initial attempt to adapt an existing model to this end.

The problem environment, the IRS modernization project, has spanned three decades and has cost over $14 Billion in its various incarnations (see, for example, Varon 2005). The project has been revamped, stopped and restarted several times, reflecting a continuing problem that until recently appeared intractable (Holmes 2006). The problem environment is reminiscent of well-behaved and simple concept models of projects (Richardson and Pugh 1981) and (Sterman 2000) with three new structures: The influence of "scope creep," a pressure to add functionality to the project, a sector that captures management's perceptions of the project's viability, and decision rules for project termination and restarting. As anticipated, adding changes to the requirements set increases the resources needed during the life of the project. This in turn triggered the decision rule that caused premature termination.

Building on previous work (Rich et al. 2007), the ultimate goal of this research stream is to integrate descriptive research of escalating projects with the feedback based insights available from dynamic modeling. This combination may provide understanding of the effects of firm capabilities, policy changes, and scheduling conflicts on project success. Working with simple project models is the first step towards an elaborated model, grounded through retrospective analysis of project materials, to learn more about IT development in organizations under stress.

Background

System dynamics has been used by a number of researchers to further understanding of the behavior of projects. Studies of project dynamics often focus on unrecognized misconceptions of project progress and subsequent rework cycles as the key to understanding budget and schedule overruns (Cooper 1980; Reichelt and Lyneis 1999). Application of this perspective to the specific issues of software has been documented and extensively tested by Abdel-Hamid and Madnick (1990, 1991). Their ideas have been disseminated widely in the software development literature, with several hundred citations since its original publication two decades ago[1]. Their work is quite well developed, but is challenging for those new to dynamic modeling. Therefore we chose to begin this analysis with an illustrative approach using a simpler project model, looking for a broad conceptual understanding.

The IRS efforts at modernization provide a good context for studying large-scale IT projects in the public sector for several reasons. First, the IRS is considered to be a very well-managed agency by IT researchers (Bozeman 2002) and if such a project were to be successful anywhere, one might expect that to happen at the IRS. Second, the IRS made four distinct major attempts to modernize since the late 1960s (see Appendix 1). The first three attempts clearly failed and were abandoned at a combined cost over $4 billion (in nominal value): the Tax Administration System Project from 1969 to 1977; the Service Center Replacement System project from 1978 to 1986; and the Tax Systems Modernization project from 1987 to 1997. Along the way, there were over 40 other projects aimed at initiating or furthering agency modernization, most of which were also abandoned as failures (*ibid*) The fourth attempt, the Business Systems Modernization Project, began with the Blueprint Project around 1998 and is still underway with an estimated cost in excess of $10 billion (Varon 2004). While experiencing some success, the current attempt is already significantly over budget and behind schedule (Johnston 2003). Third, the various iterations of the IRS modernization effort received wide-scale attention in the media, in oversight organizations, and within the agency itself. The ease in acquiring longitudinal historical data on the project at a relatively low cost, and the nature of the

[1] A recent Google Scholar search found almost 400 references to the Abdel-Hamid and Madnick text.

IT project itself, made this an ideal case to study to learn more about abandonment processes in large-scale public-sector IT projects.

Problem Context

The Internal Revenue Service received 228 million citizen and corporate tax returns in 2006, somewhat more than the 202 million received in 1995 (U.S. Department of The Treasury 2006, 1995). Taxes on personal incomes, including social insurance programs, represent the bulk of resulting revenue. Along with population pressure, this growth reflects the increasing complexity of the tax code, as tax brackets have been modified and new programs implemented to capture income streams for governmental operations.

The current information systems supporting the IRS date back to the late 1950's, and reflected the architecture and data processing methods available at that time. Much of that architecture is still in place, relying on obsolete hardware and software of vertigo-inducing complexity. Each modernization effort was pledged to untangle the code and bring modern approaches and technology to bear.

As is often the case, the IRS modernization efforts are deeply enmeshed in a changing context that is not in their control. Congressional and executive branch policymakers introduce innovations that reflect their concerns about the current and future state of the economy. Changes to tax laws and modification in enforcement priorities engender alterations to the same software systems that are undergoing modernization. The wrapping of new policy requirements with those needed to replicate the processing already in place greatly increases the complexity of change. Even the simplification of the tax code during 1980s did not result in compliance savings from the taxpayer perspective (Slemrod 1992) or the IRS, as the concurrent presence of old and new tax laws in the same system was required.

The changing policy and oversight processes to which the IRS modernization efforts are subjected help produce one of the familiar banes of project management: scope creep. Scope creep, or the altering or adding of requirements to an ongoing project, has been shown to adversely affect the likelihood of a project finishing on time and on budget (Lamberti and Wallace 1990). Scope creep can arise through factors such as poor requirements definition and is often addressed by a change control board that establishes change policies and the approval process for those changes (Wiegers 2003). However, the IRS modernization efforts differ from private sector IT projects in terms of the nature and source of the scope creep. While project managers employed by private corporations perceive requirements and scope risks as highly controllable through good project management (Lamberti and Wallace 1990), public organizations, like the IRS, often have requirements changed or new requirements introduced exogenously from higher level federal bodies or changing legislation. Under these conditions, project managers at the IRS will perceive the scope creep problem as uncontrollable which will further increase the project's execution complexity and difficulty (Lamberti and Wallace 1990) We argue that the deleterious effects of repeated, uncontrollable scope creep combined with an already complex project position the IRS for failure in its attempts to modernize.

As a first step, we consider one of the areas of complexity that arose in the IRS projects: The need to implement changes to tax policy in software while attempting to replace existing applications. This can be phrased as a simple dynamic hypothesis the effects of changing requirements on the characteristics of project success.

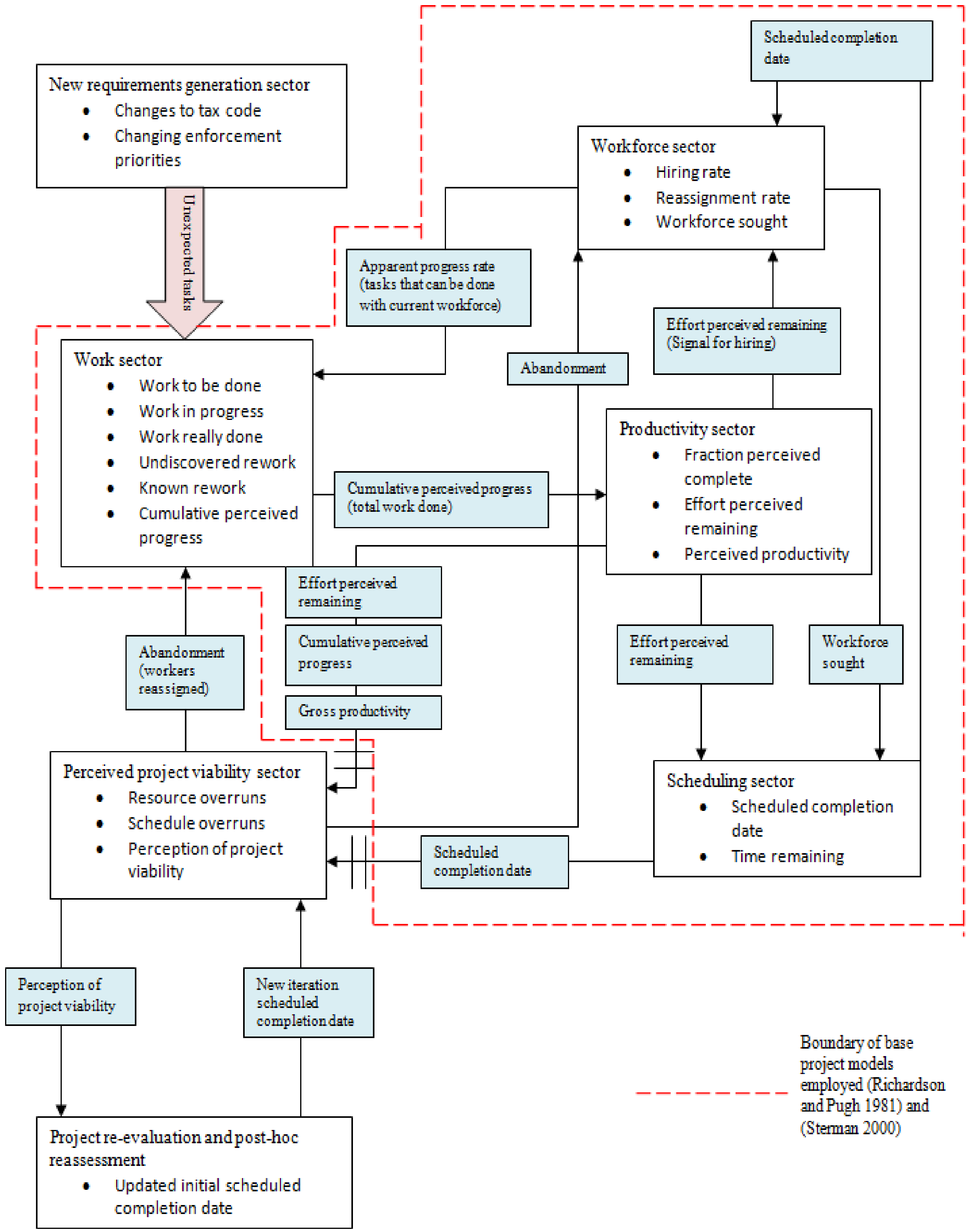
> *Hypothesis: Increasing the number of user requirements introduced during the progress of an IT project will increase resource use and likelihood of project termination.*

This hypothesis combines the insights from project management models about project outcomes (Cooper 1980; Abdel-Hamid and Madnick 1990, 1991) with the provision of information about the future of the project. This provides a linkage to thinking about project escalation and de-escalation.

The construction of a metric for success or failure is a necessary linkage between the project models and the oversight environment. We would expect to find that even successful projects run into risky periods and persevere through to completion, while unsuccessful projects find themselves in situations where risk compounds without resolution. While efforts are being made to predict project complexity at its beginning (Xia and Lee 2004), the effects of complexity often emerge during the details of software design and integration. Projects may well tip into a failure mode where insiders recognize the inevitable, but the news has not reached their managerial or oversight counterparts.

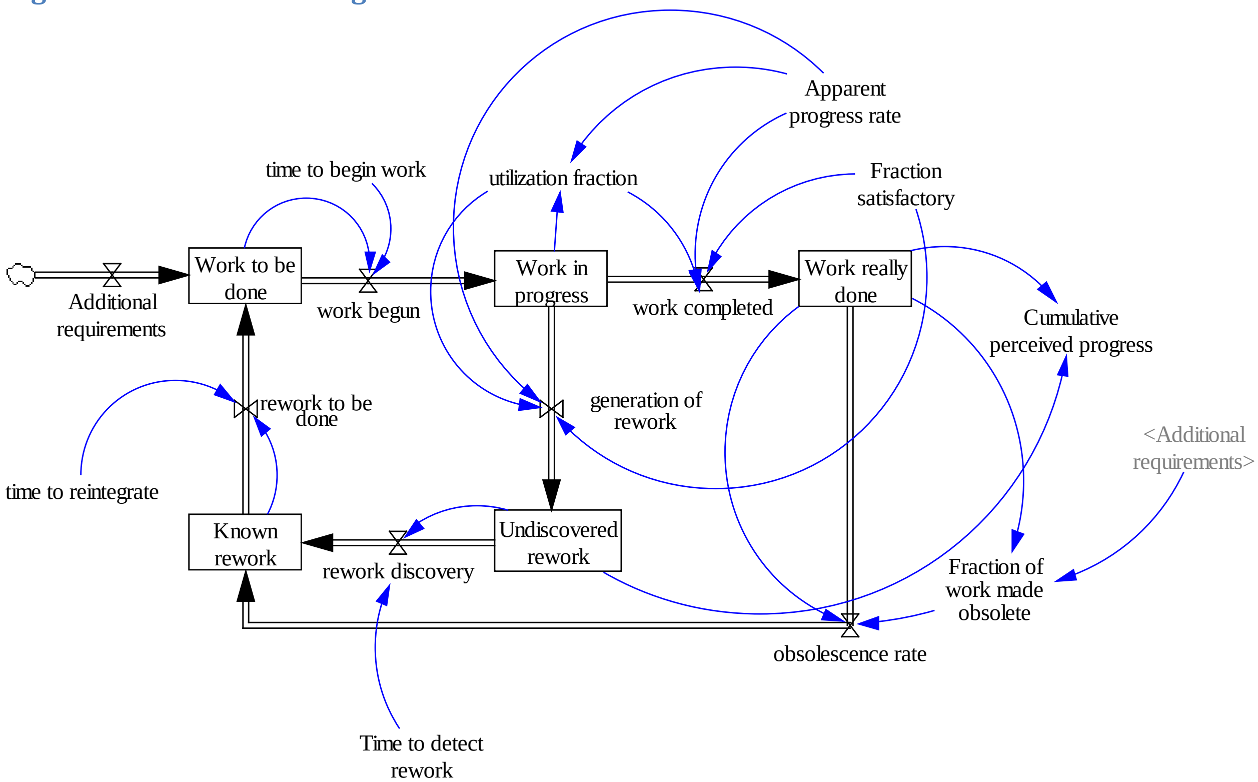
The definition of organizational tolerance for failure is not cut and dry, as it is quite possible to imagine situations where organizations see the slim chance of project success better than the consequence of stopping, or where the achievement of substantial portions of the work is considered sufficient for its purpose. For this work we assume that a project that reaches 80% of the established requirements within 150% of the initial timetable or 150% of the initial resource estimate (in staff hours) is successful. Thus, perceptions of project success in the model, and hence decisions to abandon a project, depend on cost and schedule overruns. There is some debate among scholars as to whether cost and schedule overruns are a cause of project abandonment (Lyytinen & Hirschheim, 1987) or whether they are symptoms of the underlying project problems and serve to send a signal to senior management about the declining state of the project (Ewusi-Mensah, 1997). Whether they are the root causes or function as cues that signal information about project status, the use of these factors as decision criteria seems justifiable. The next section provides a general overview of the structure of the simulation model followed by detailed descriptions of model sectors. The model (see Figure 1) consists of seven total sectors, four of which are drawn from the base project model and three are added to adapt the model to the case study. The boundary of the original project model is demarcated by the dashed red line.

Figure 1: Model Structure



Work Sector

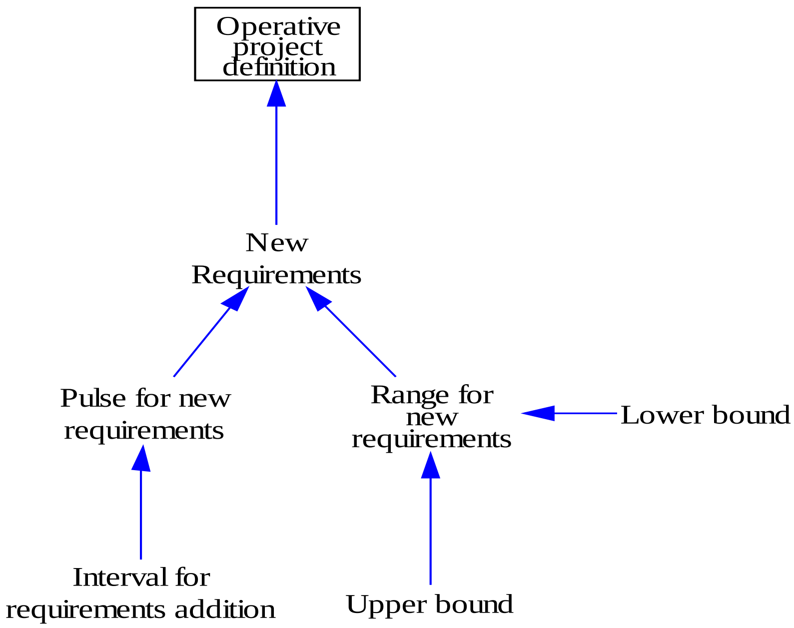
Figure 2: Work Sector Diagram



"Work to be done" starts out with the value of the initial number of tasks in the original project definition. If additional requirements are being introduced, they initially enter the stock of "Work to be done" as well. Assuming that the project has not been abandoned (variable "Work switch", see formula for "Work begun") then the tasks that have been identified will be sent to be worked on. "Apparent progress rate" represents the possible amount of work that can be done with the current number of employees and the current productivity. If the amount of "Work in progress" exceeds the amount of work that can be done in the next month, then the "utilization fraction" will be equal to one. If the amount of work that can be done exceeds the amount of work that needs to be done, then "utilization fraction" will be adjusted so that "Work in progress" will not go negative. In addition, the amount of work completed depends on the quality of the work, represented here as "Fraction satisfactory". A certain amount of work being done is not done correctly which results in the "generation of rework". Work that is done to specifications is represented in the stock, "Work really done". Work that has been done correctly can be made obsolete by the introduction of additional requirements. Since it is conceivable that in the early days of the project the number of new requirements introduced could exceed the amount of work completed successfully, the "obsolescence rate" is defined as the "Faction of work made obsolete" multiplied by the "Work really done". "Fraction of the work made obsolete" is constrained so that it cannot exceed one. The "generation of rework" accumulates in "Undiscovered rework", or work done incorrectly but not known to be so. Together "Work really done" and "Undiscovered rework" constitute "Cumulative perceived progress". In time this flawed work will be discovered. The "Time to detect rework" depends on the stage the project is in. If the project is perceived as mostly complete, there will be more individuals put into testing which will reduce the "Time to detect rework". This discovered rework along with the completed work that is made obsolete accumulates in "Known rework". This "Known rework" is then reintegrated into the planned "Work to be done".

New Requirements Generation Sector

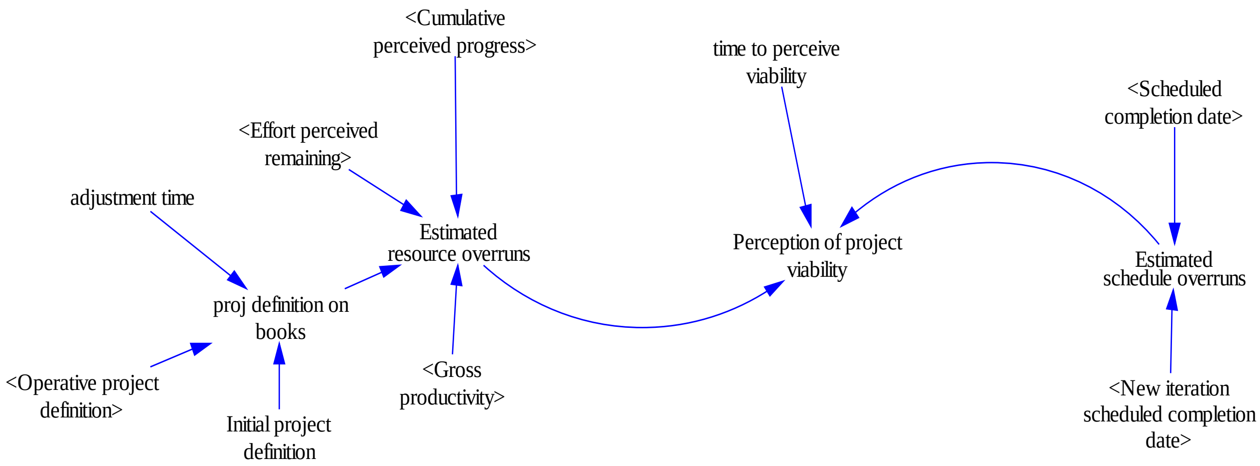
Figure 3: Requirements Generation Sector Diagram



This sector represents the generation of additional requirements. In the IRS case, Congress and the Whitehouse are responsible for making changes to the tax codes and to enforcement priorities that result in additional requirements. "Operative project definition" includes the initial number of tasks to be performed plus any additional tasks generated in "New Requirements". "New Requirements" is a random number of new requirements within a specified range ("Range for new requirements") pulsed in ("Pulse for new requirements") at a random interval ("Interval for requirements addition").

Perceived Project Viability Sector

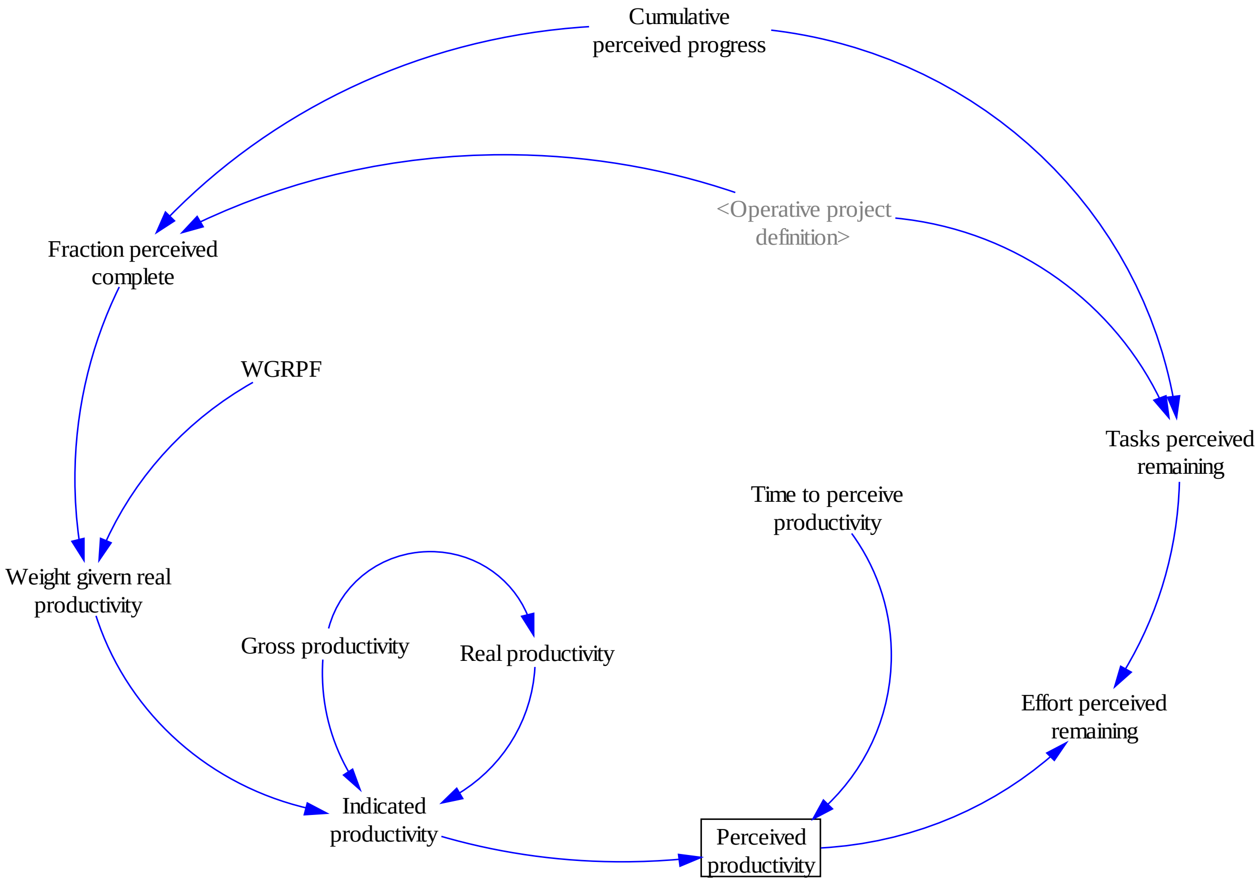
Figure 4: Perceived Project Viability Sector Diagram



The purpose of this sector is to simulate how project managers come to conclusions about the viability of a project that is underway. "Perception of project viability" is a nonlinear function of "Estimated schedule overruns" and "Estimated resource overruns". "Estimated schedule overruns" is the ratio of the scheduled completion date to the initial scheduled completion date for the project (or that instance of the project if there is abandonment and restarting of projects). "Estimated resource overruns" is the ratio of the work already done plus the work that needs to be done to the original amount of work that was anticipated. The amount of work that is anticipated is the "proj definition on books". "Proj definition on books" is the initial number of tasks originally required for the project plus any new requirements generated by Congress or the White House smoothed by some time to update the project requirements on the books ("adjustment time").

Productivity Sector

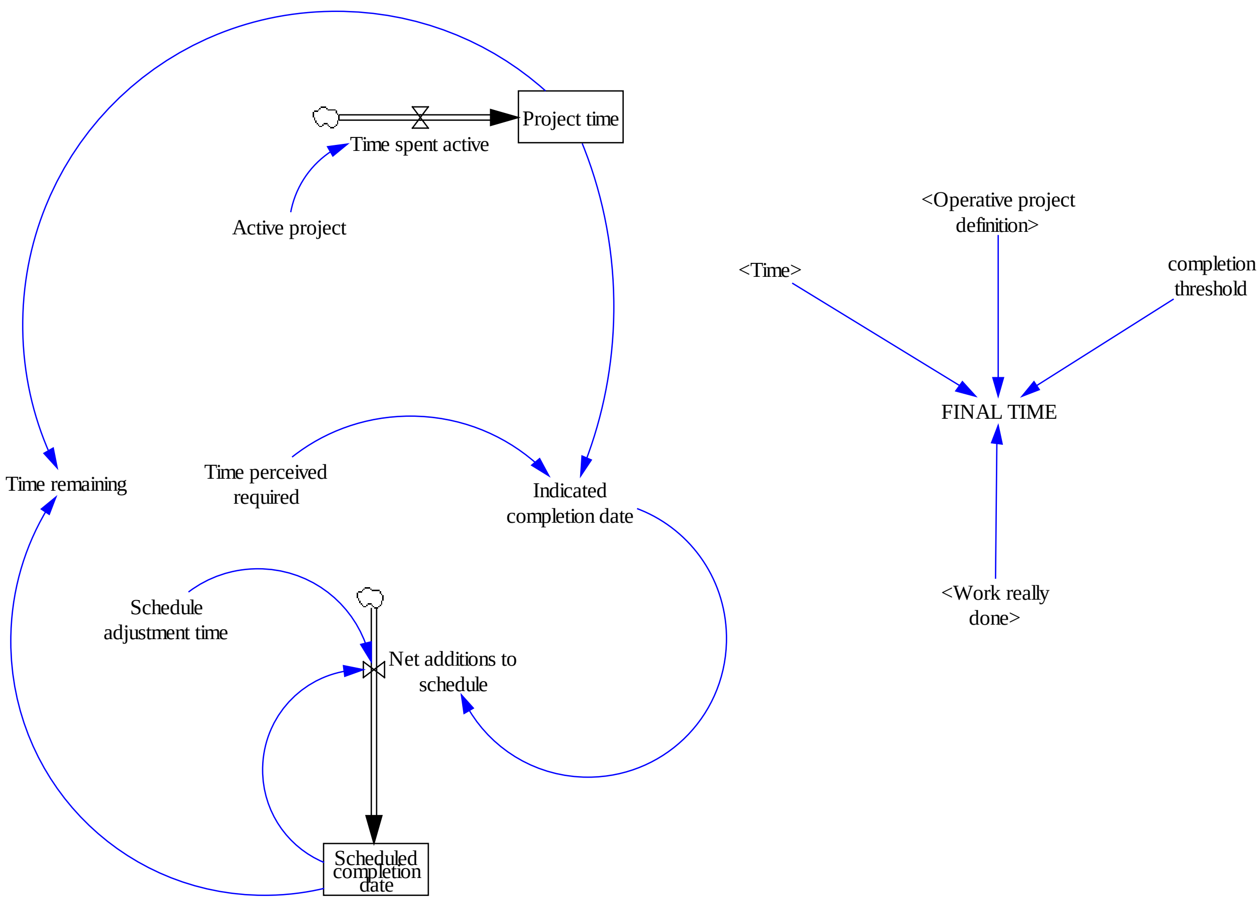
Figure 5: Productivity Sector Diagram



This sector describes the changing focus from total productivity to real productivity as the project nears completion. "Fraction perceived complete" is the ratio of the "Cumulative perceived progress" to the "Operative project definition". As the project nears completion and workers begin to think about testing, project managers begin thinking more about "Real productivity" than "Gross productivity". Thus, the "Weight given real productivity" increases. "Real productivity" is the fraction of the work that is completed that is done correctly ("Fraction complete"). Thus "Indicated productivity" will decrease as managers start thinking less about the total fraction of work done and more about the fraction of work done right. "Perceived productivity" is a smoothed version of "Indicated productivity" that incorporates the reality that there is a delay in perceiving the instantaneous changes in "Indicated productivity". As "perceived productivity" decreases the "Effort perceived remaining" increases. "Tasks perceived remaining" is merely the difference between the total number of tasks completed and the total number of tasks to complete. As "Tasks perceived remaining" increases, "Effort perceived remaining" increases.

Scheduling Sector

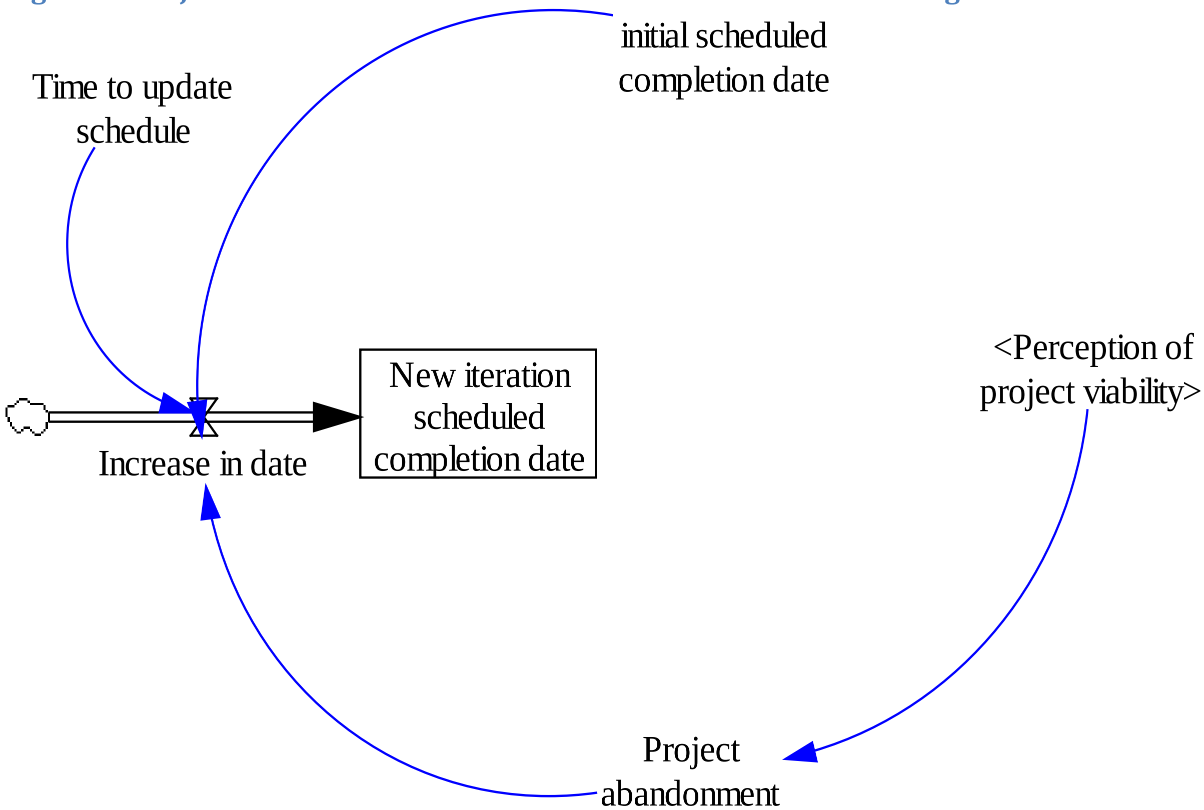
Figure 6: Scheduling Sector Diagram



This sector represents the process of changing the project schedule once the project is underway. If the project has not completed, the "Scheduled completion date" of the project is updated by "Net additions to schedule". "Net additions to schedule" reflects the difference between the current "Scheduled completion date" and the "Indicated completion date". The "Indicated completion date" is derived from the time already spent on the project, "Project time", and the projected amount of time left on the project, "Time perceived required". "Time perceived required" is determined by dividing the number of man-hours (or in this case man-months) left by the number of workers you intend to have in your workforce. "Project time" is the amount of time that the project has been experiencing active work ("Active project"). If there is no work being done on the project because it is in a state of abandonment then "Project time" remains constant. "Time remaining" is the difference between the number of months that the project is forecast to finish in less the time the project has spent in activity. When the project is completed, "FINAL TIME" is equal to the current time and the run ceases. Otherwise, "FINAL TIME" is set to 360 months.

Project Re-evaluation and Post-hoc Reassessment Sector

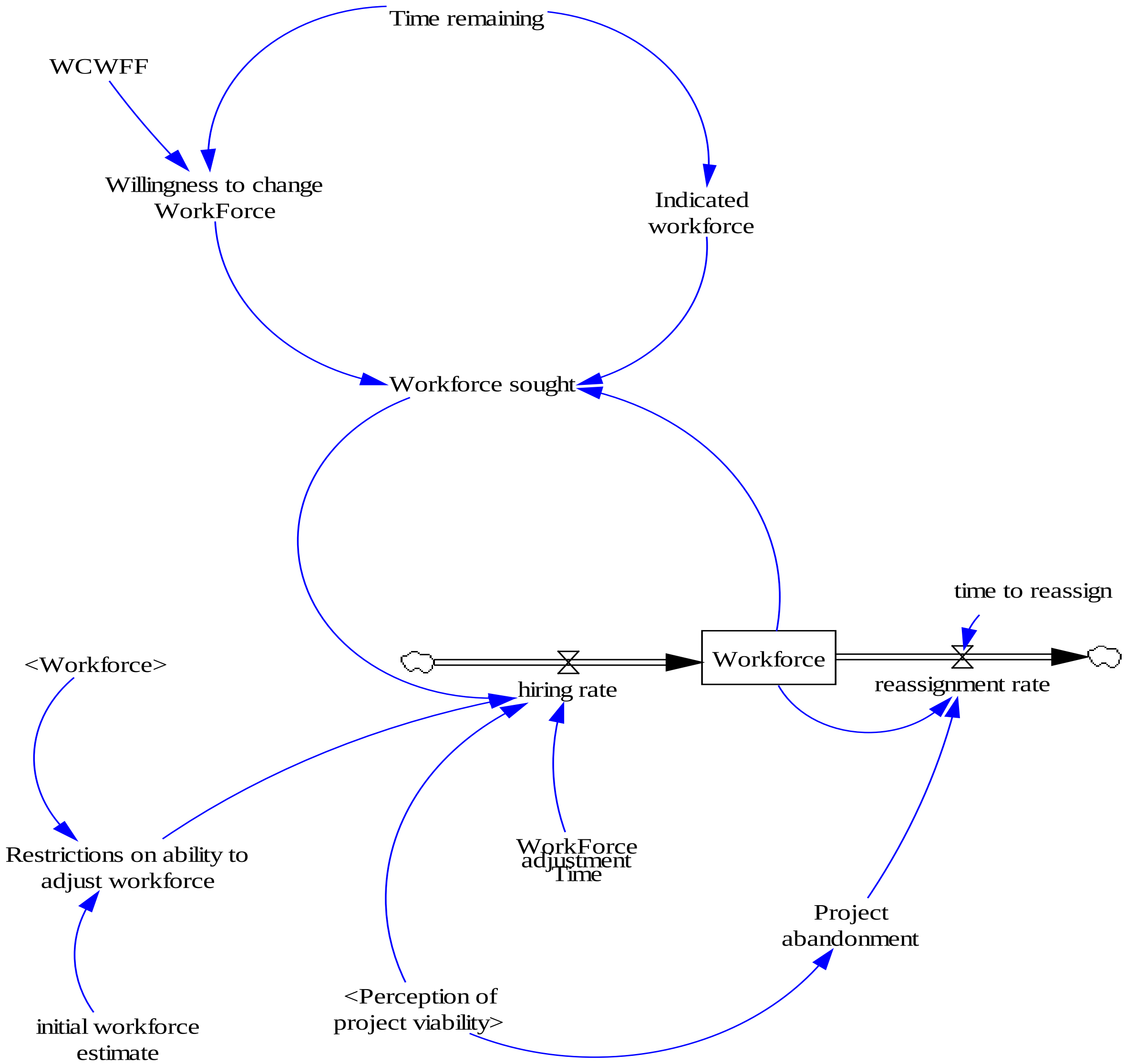
Figure 7: Project Re-evaluation and Post-hoc Reassessment Sector Diagram



This sector describes the manner in which the scheduled completion date for the next iteration of the project is updated after the current iteration is abandoned. Once the "Perception of project viability" reaches a certain low level, it triggers "Project abandonment". Once the project has been abandoned a scheduled completion date for the new iteration is set ("New iteration scheduled completion date"). This new date will be later than the original completion date ("Increase in date"). The "New iteration scheduled completion date" for each new iteration of the project will vary depending on the length of time that the project is perceived to be not viable.

Workforce Sector

Figure 8: Workforce Sector Diagram



"Workforce" reflects the number of employees performing active work on the project. When a project is abandoned a number of employees may remain nominally assigned to the project but they will not be performing active work on the project and thus will not be included in the "Workforce". "Workforce" is increased by the "hiring rate". The "hiring rate" is derived from the "Workforce sought", "Restrictions on ability to adjust workforce" and "Perceptions of project viability". "Perceptions of project viability" are used to dampen the number of workers hired relative to those needed ("Workforce sought") and is used to describe the phenomenon that managers might be less willing to hire new workers if they have a negative opinion of the possibility for a successful project. "Restrictions on ability to adjust workforce" is a parameter put in place to limit the extreme degree of hiring flexibility that the base model assumes. We assume that a large bureaucratic organization like the IRS would be less flexible in terms of hiring or contracting additional labor than a private sector organization and thus more likely to push back the scheduled completion date rather than hire large numbers of workers. Thus, "Restrictions on ability to adjust workforce" is a function of the relationship between the current workforce and the initial workforce estimate for the project. The more the workforce exceeds the initial workforce estimate, the more difficult it will be to hire additional workers. "Workforce sought" is a weighted sum of the "Indicated workforce" and the current "Workforce". The weight is the "Willingness to change workforce" which assumes that as "Time remaining" for the project runs out, the management will be less willing to hire more workers. The workers will be reassigned ("reassignment rate") if the project has been abandoned ("Project abandonment").

Presentation and analyses of model behavior

Figure 9: Base run behavior (no additional requirements)

The graph below depicts the base run behavior which represents a successful project.

<u>Base run parameters:</u>

Initial workforce=2

Initial scheduled completion date=70

Initial project definition =5500

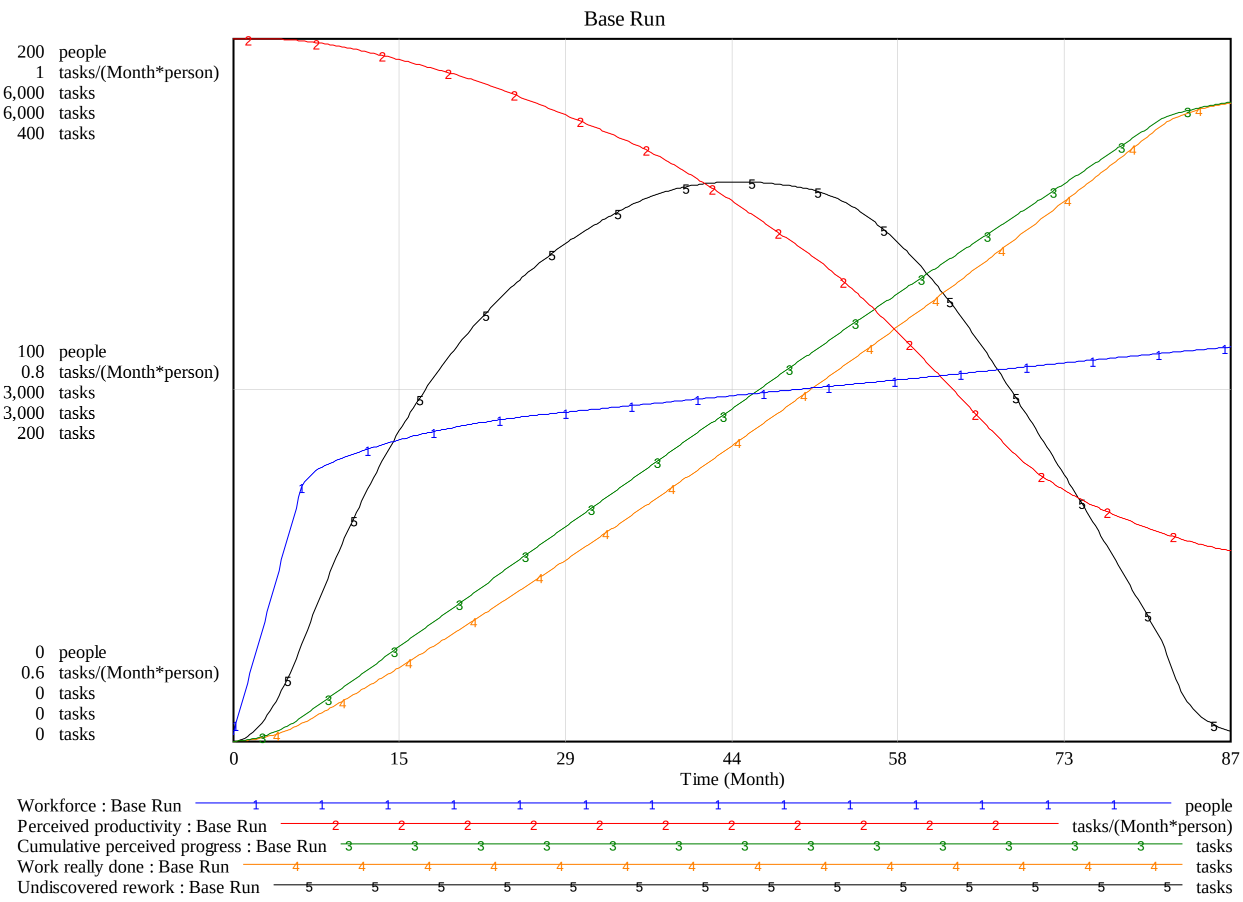
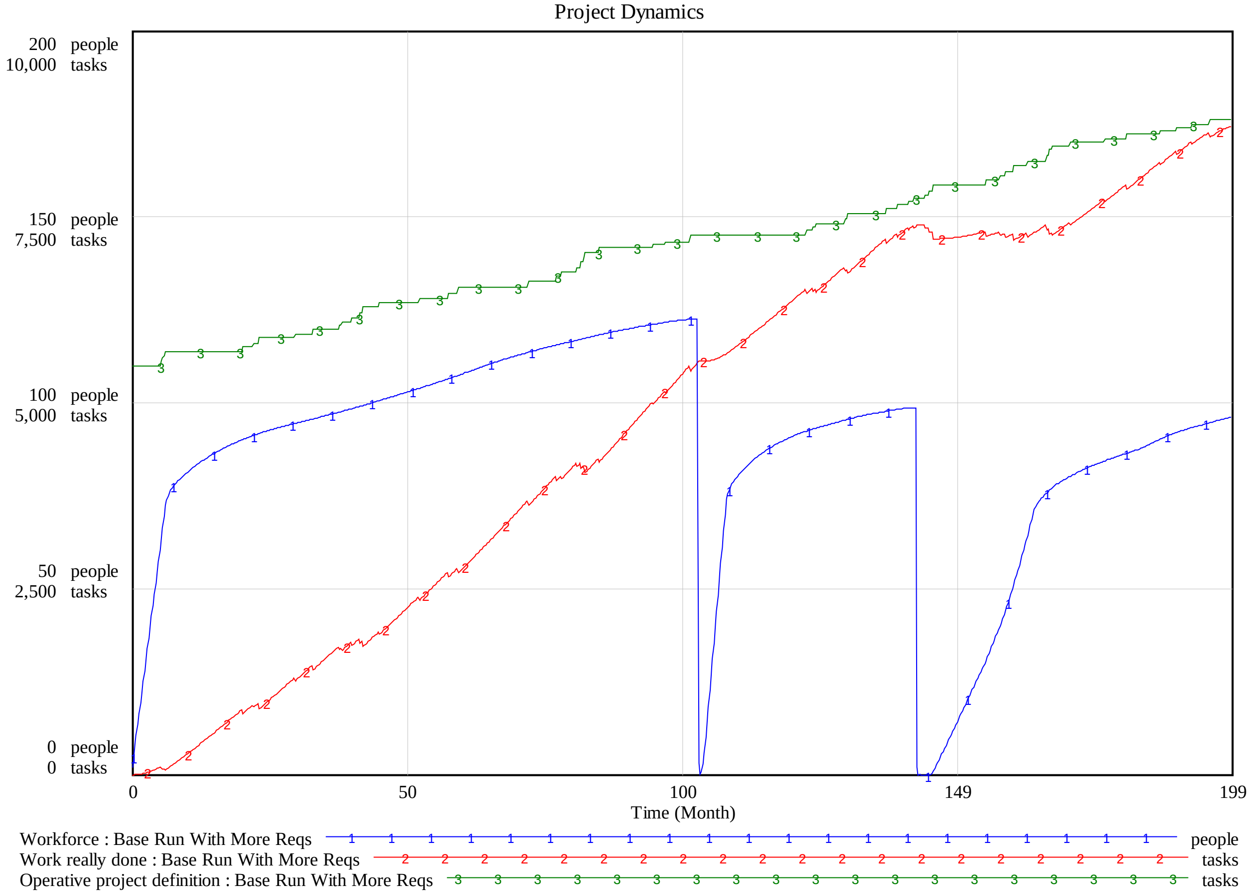


Figure 10: Base run behavior with additional requirements (Part 1)

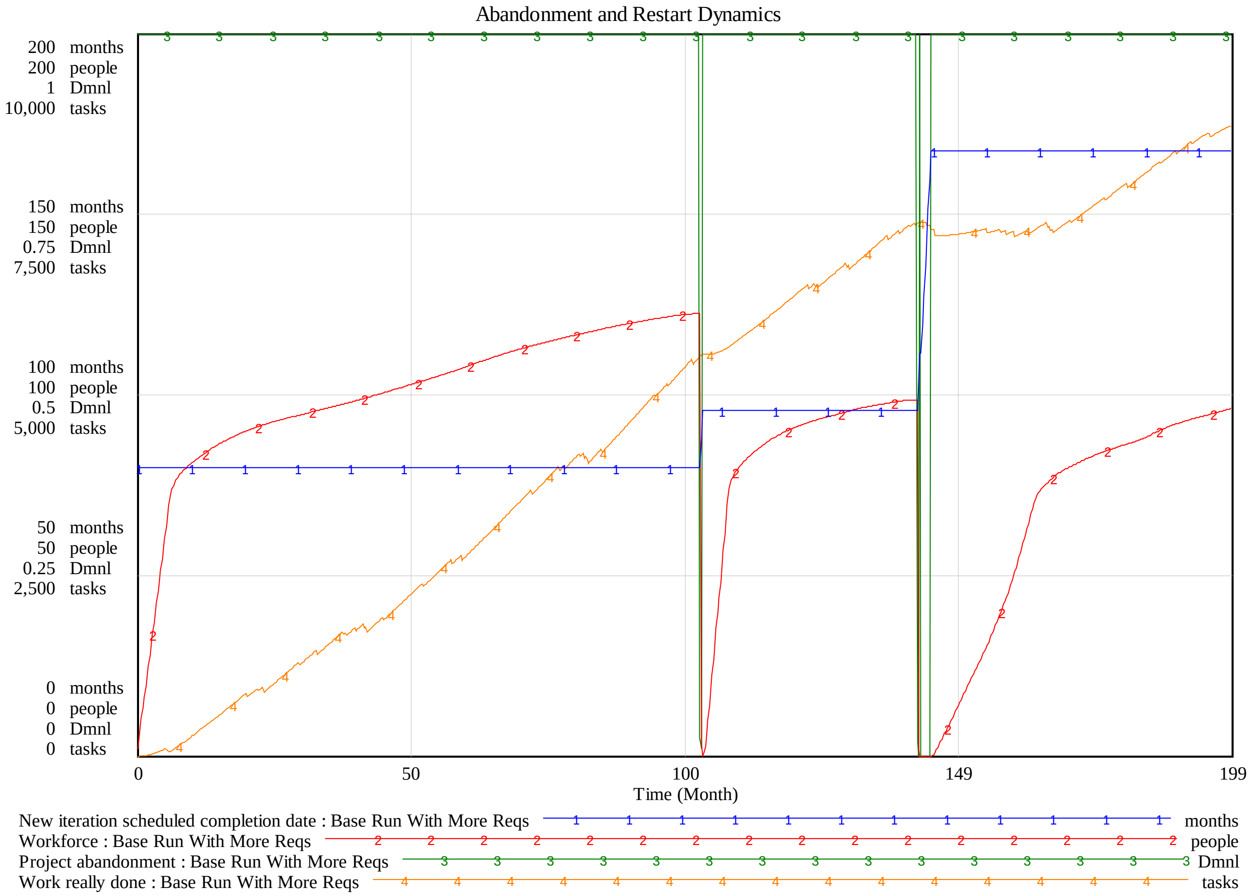
The graph below depicts the base run behavior of a project that suffers from repeated escalation and abandonment.

<u>Base run parameters:</u>
Initial workforce=2
Initial workforce estimate=70
Initial scheduled completion date=80
Initial project definition=5500



As can be seen from the model behavior above, there are three distinct iterations of the project before it finally successfully completes in month 199. The operative project definition began at 5500 tasks but rose all the way to 8803 tasks over the course of about 17 years. This increase in the operative project definition is caused by the continuous pulse of additional requirements throughout the lifecycles of the projects.

Figure 11: Base run behavior with additional requirements (Part 2)



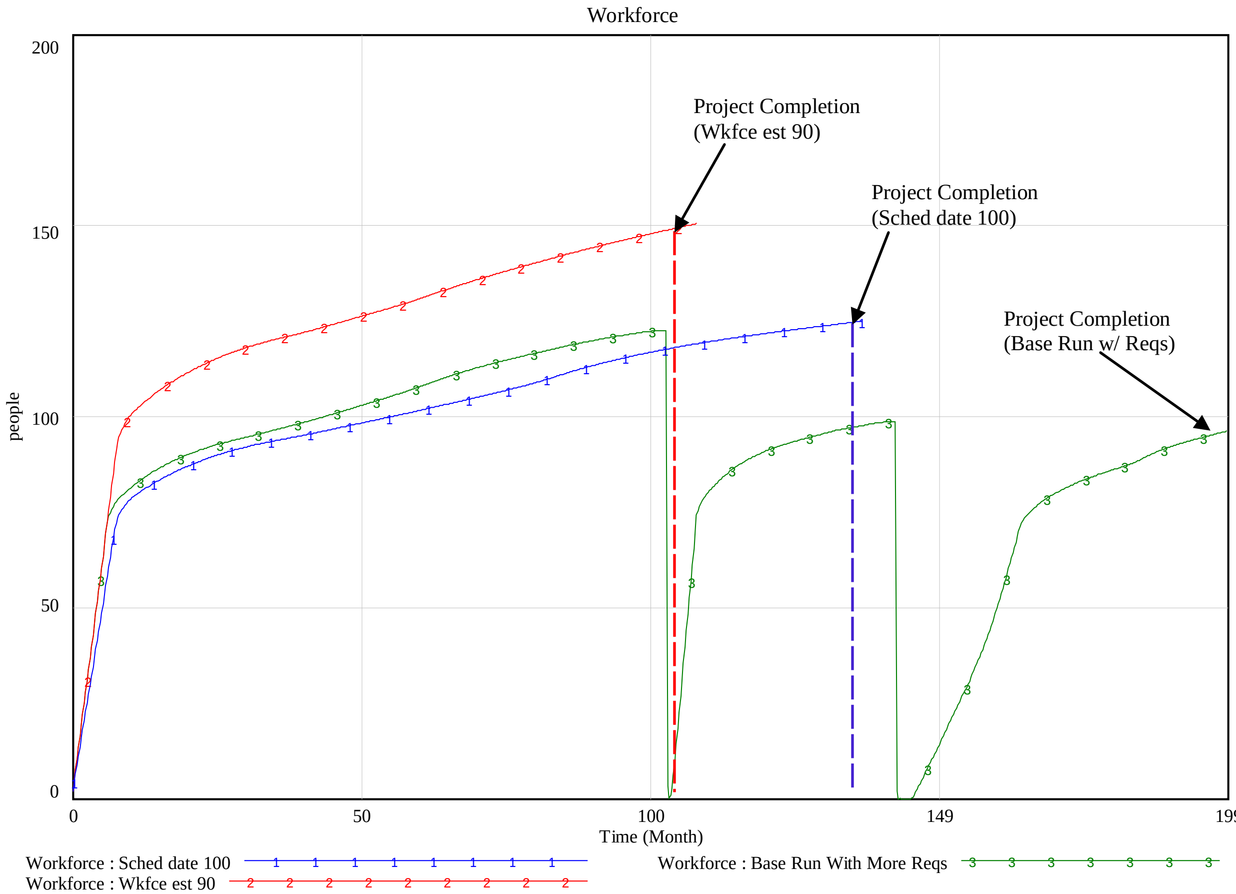
Once a project has been abandoned the scheduled completion date for the next iteration of the project is determined. As long as the project continues to be perceived as unfeasible, the scheduled completion date for the next iteration continues to increase until the next iteration is perceived as feasible and begins. This phenomenon can be seen from the "New iteration scheduled completion date" curve above. Note the difference in the degree of increase of the scheduled completion dates among iterations of the project. The less viable the project is perceived to be when it is abandoned the longer it will remain abandoned and thus the greater the increase in the initial scheduled completion date for the next iteration. This is seen in the final iteration of the project.

Policy analyses

Initial workforce and schedule estimates

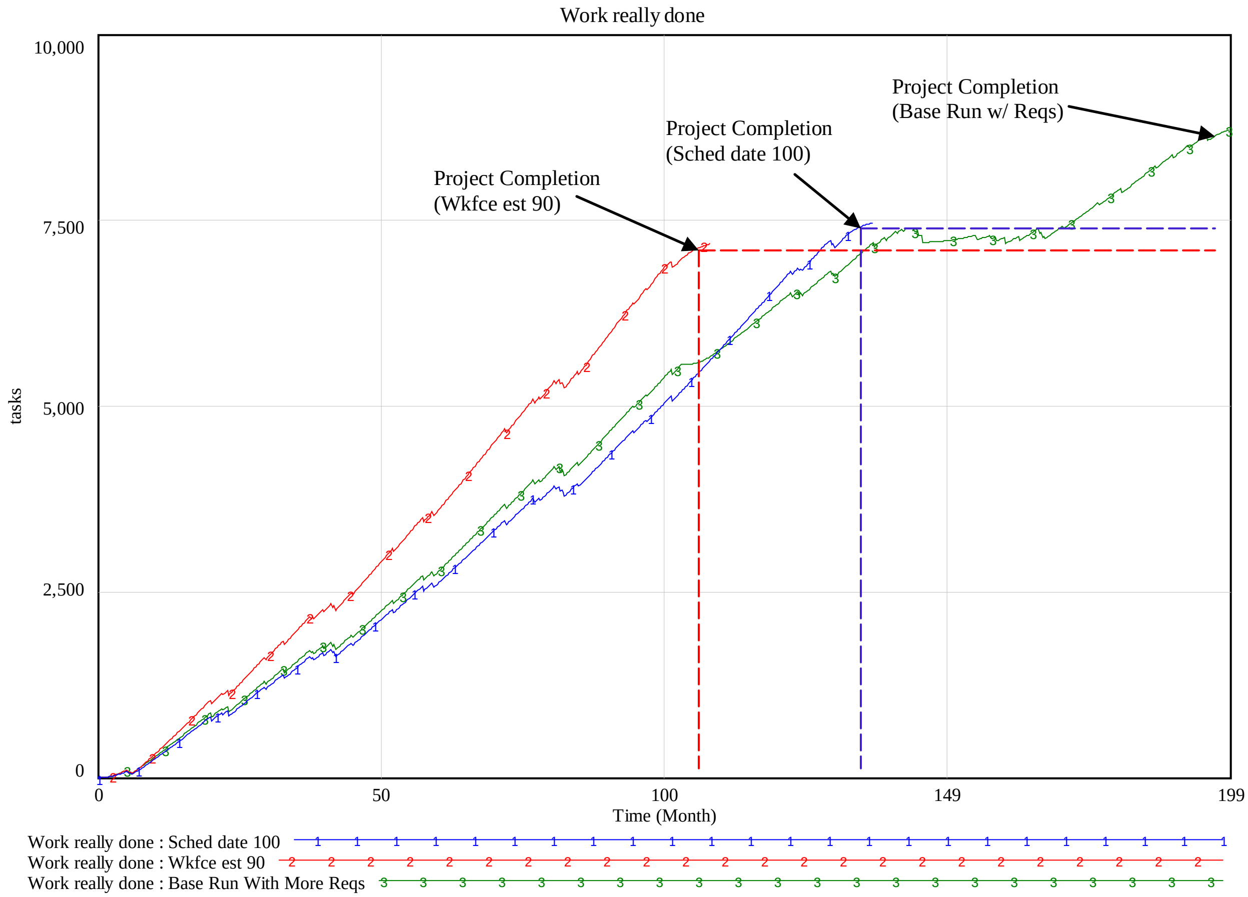
As can be seen from the graph below, "initial workforce estimate" and "initial scheduled completion date" are obvious policy parameters.

Figure 12: Policy Option: Workforce and Schedule Padding (Part 1)



This graph suggests that if sufficient padding had been built into either the scheduled completion date or the projected number of workers then the project could be capable of finishing in a single iteration. These two policy options also reduce the total amount of work that needs to be done to complete the project.

Figure 13: Policy Option: Workforce and Schedule Padding (Part 2)



Even though there is some room to use these two parameters as policy options, it is obviously unrealistic to forecast enormous human and time resources so that the project stays under budget and schedule and therefore doesn't meet abandonment criteria.

Training

One of the commonly suggested policies to mitigate project failure is training employees. This was actually done in the IRS modernization effort between the third and the uncompleted fourth iterations of the project. The model shows that training and thereby increasing "Fraction satisfactory" and thus "Real productivity" can have a dramatic impact on the completion of a project. Raising the "Fraction satisfactory" from .7 to .85 allowed the project to finish 93 months earlier and with only 78% as much work as the base run.

Figure 12: Policy Option: Training (Part 1)

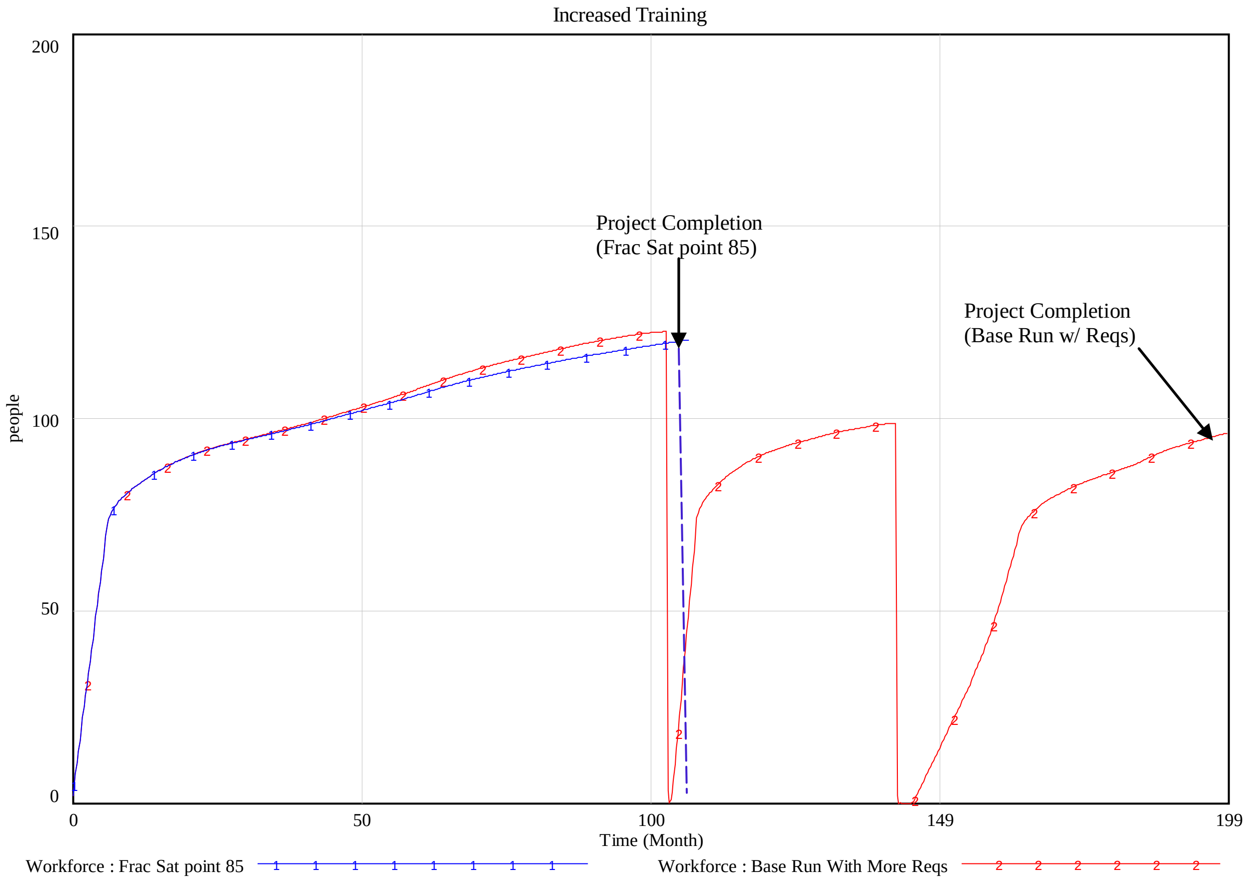
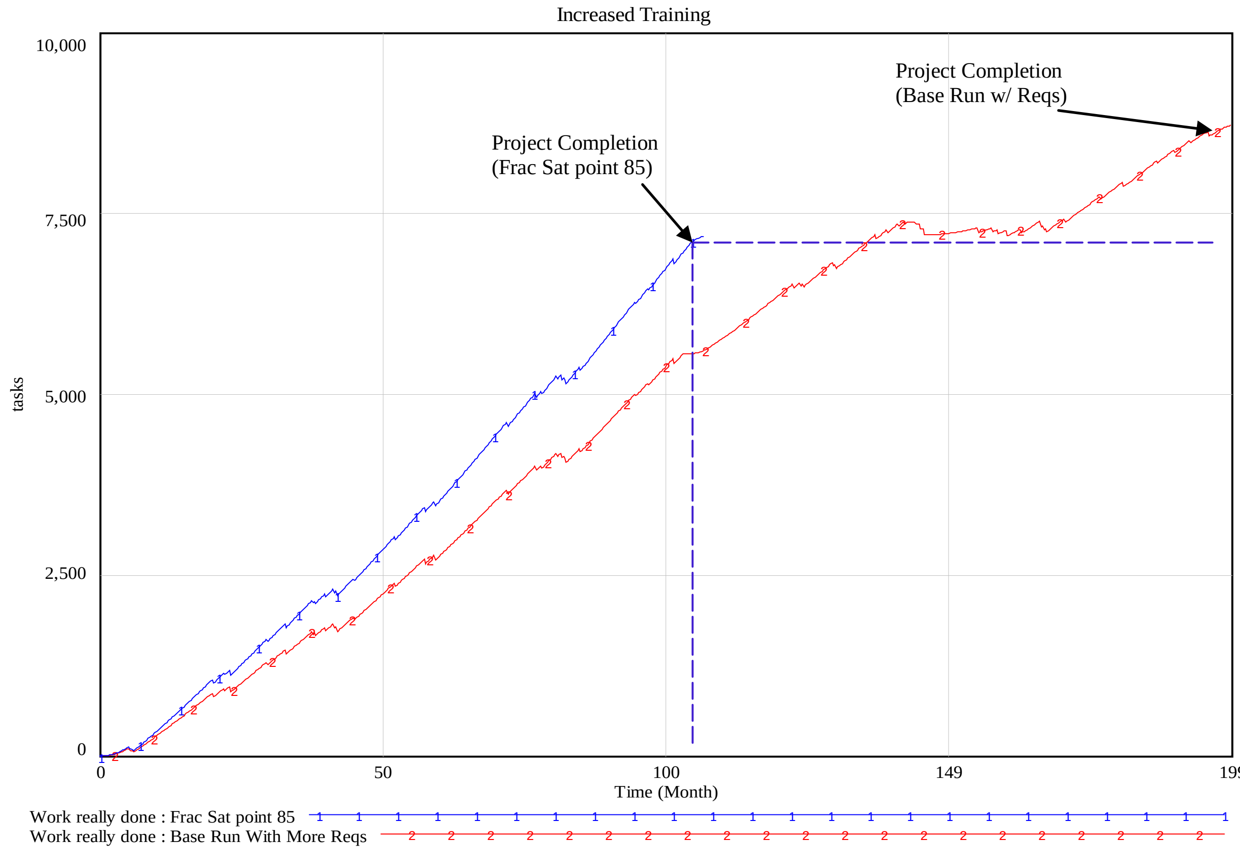


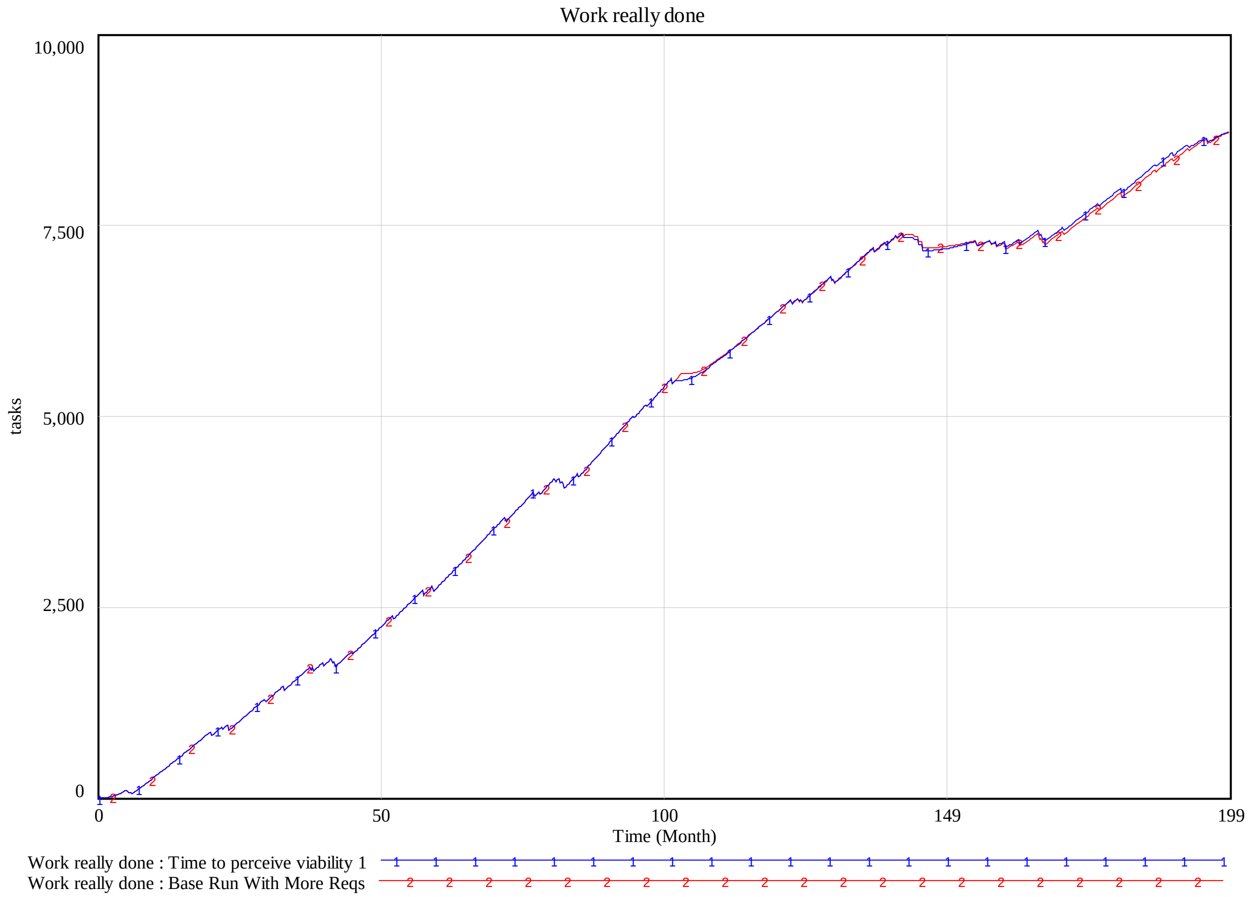
Figure 13: Policy Option: Training (Part 2)



Monitoring the project more closely

Another commonly suggested approach to improving project performance, especially in the project management literature, is to improve monitoring of the project as it is underway. The model suggests that if this monitoring reduces the "time to perceive viability" of the project, it will confer little benefit. However, this result makes the arguably unrealistic assumption that while a project manager has been given a greater ability to monitor project progress, he does not have any mechanism to ameliorate the poor progress of the project that he is more acutely noticing. Some possible actions a project manager could take under these conditions are to bring in outside consultants or to reassess the project plan while keeping the project active. Thus, the more appropriate claim here is that absent an effective plan to deal with unfavorable results, monitoring the project more closely will not improve performance.

Figure 14: Policy Option: Monitoring the project more closely



Conclusions

The repeated cycles of project escalation and abandonment at the IRS have resulted in an IT modernization effort that has spanned 30 years and cost taxpayers billions of dollars. We present a dynamic hypothesis that these cycles of escalation and abandonment are the result of the exogenous introduction of additional requirements during the lifecycle of the project. A simulation model was constructed to test the hypothesis and the results support the proposition that scope creep resulting from the introduction of new requirements may be a causal factor in the cycles of project escalation and de-escalation. The policy options tested in the simulations indicate that incorporating moderate padding into the initial project workforce estimate and training employees both mitigate project escalation and abandonment. Thus, we would recommend these actions be taken to reduce the possible negative effects of scope creep. The model did not provide support for the conjectures that monitoring the project progress more closely and padding the initial schedule estimate will significantly mitigate project escalation and abandonment.

Limitations

There are some limitations that are important to note. We need to be sure that we identify and address the problems seen in the IRS data, not just those that are present in the concept model. There are other alternative published formulations (e.g., those developed in Abdel-Hamid and Madnick 1991) that might well serve as guideposts for modeling repeated project failures. The choice of concept model can prematurely impose an inappropriate conceptual lens on the problem. Additionally we need to perform a thorough content analysis of the historical documents we have collected in order to derive reference modes for key variables which will allow us to evaluate and support the model. Finally, the model has certain components that are most likely overly discrete, namely the decision mechanisms around project viability and termination as well as the complete reassignment of the workforce after project abandonment. As we continue to develop the model and calibrate it to reference modes gleaned from the historical documents we hope to resolve these current limitations.

References

Abdel-Hamid, Tarek K., and Stuart Madnick. 1990. The elusive silver lining: How we fail to learn from software development failures. *Sloan Management Review* (Fall).

———. 1991. *Software Project Dynamics: An Integrated Approach, Prentice-Hall Software Series*. Upper Saddle River, NJ: Prentice-Hall.

Bozeman, B. 2002. Government Management of Information Mega-Technology: Lessons from the Internal Revenue Service's Tax Systems Modernization. In *New Ways to Manage Series*. Arlington VA: The PricewaterhouseCoopers Endowment for The Business of Government.

Cooper, Kenneth G. 1980. Naval Ship Production: A Claim Settled and a Framework Built. *Interfaces* 10 (6):20-36.

Ewusi-Mensah, K. 1997. Critical Issues in Abandoned Information Systems Development Projects: What is it about IS development projects that makes them susceptible to cancellations? *Communications of the ACM, 40*(9): 74-80.

Holmes, Allan. 2006. Baby Steps for IRS Upgrades. *CIO Magazine*, April 1.

Johnston, D. C.. 2003. At I.R.S., a Systems Update Gone Awry. *New York Times*, Dec. 11, C1.

Lamberti, D. , and W. Wallace. 1990. Intelligent Interface Design: An Empirical Assessment of Knowledge Presentation in Expert Systems. *MIS Quarterly* 14 (3):279-311.

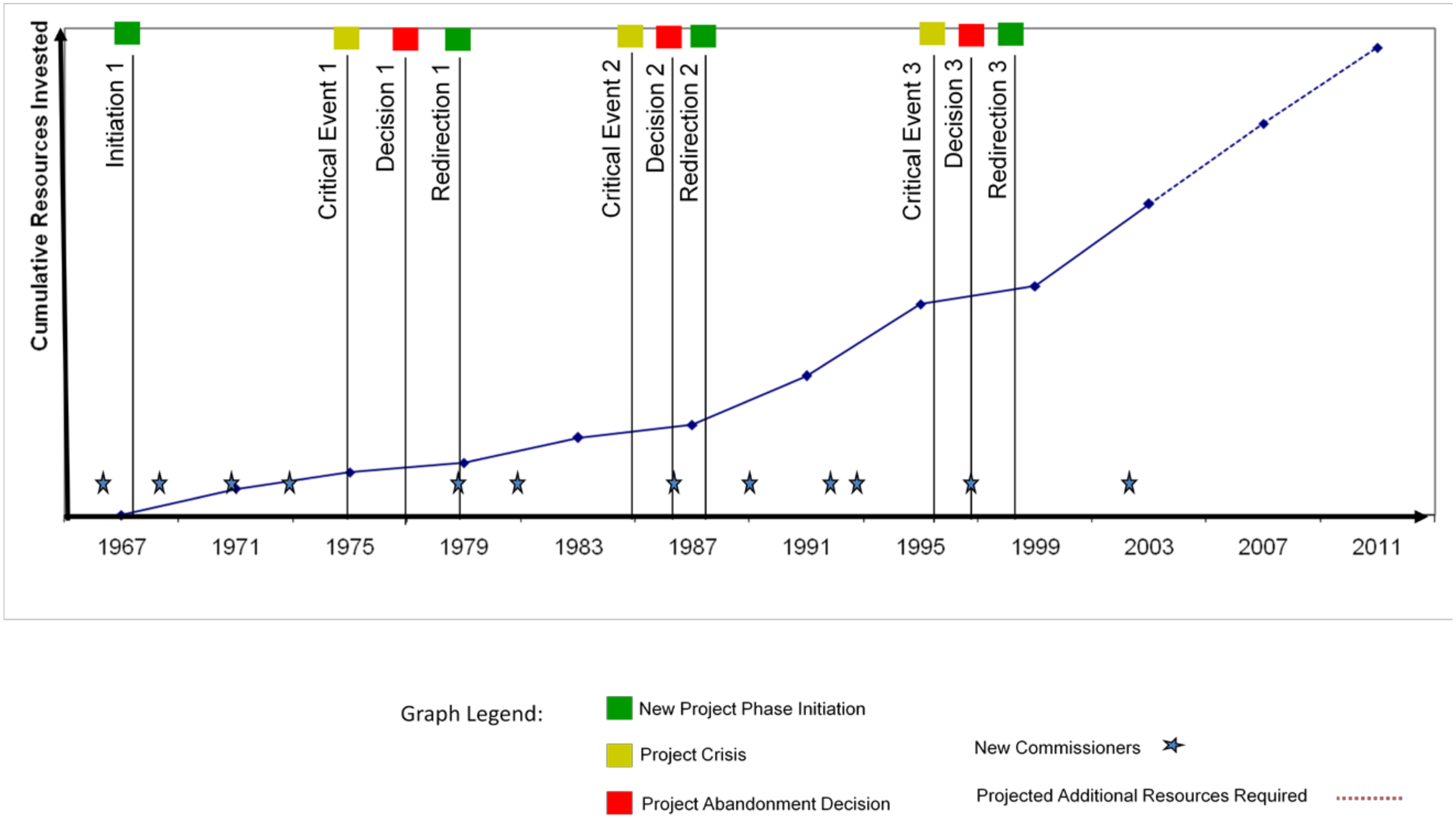
Lyytinen, K., and R. Hirschheim. 1987. lnformation Systems Failures-A Survey and Classification of the Empirical Literature. *Oxford Surveys in lnformation Technology* (4): 257-309.

Reichelt, Kimberly S., and James M. Lyneis. 1999. The Dynamics of Project Performance : Benchmarking the Drivers of Cost and Schedule Overrun. *European Management Journal* 17 (2):135-150.

Rich, E., M. Nelson, A. Whitmore. 2007. IT Project Management, Concept Modeling, and Blind Dates. *Proceedings of the 25th International Conference of the System Dynamics Society*, Boston, MA.

Richardson, George, and David F. Andersen. 1995. Teamwork in group model building. *System Dynamics Review* 11 (2):113-138.
Richardson, George, and Alexander Pugh. 1981. *Introduction to system dynamics modeling with DYNAMO*. Cambridge, MA: MIT Press.
Slemrod, Joel. 1992. Did the Tax Reform Act of 1986 Simplify Tax Matters? *Journal of Economic Perspectives* 6 (1):45-57.
Sterman, J. 2000. *Business Dynamics: Systems Thinking and Modeling for a Complex World*. New York, NY: Irwin Mcgraw-Hill.
U.S. Department of The Treasury. 1995. Internal Revenue Service Annual Data Book, edited by Department of the Treasury: U. S. Internal Revenue Service,.
———. 2006. Internal Revenue Service Data Book, edited by Department of the Treasury: U. S. Internal Revenue Service,.
Varon, Elena. 2004. For the IRS There's No EZ Fix. *CIO Magazine*, April 1.
———. 2005. The IRS Makes Progress. *CIO Magazine*, May 1.
Wiegers, K. E. 2003. *Software Requirements*. Redmond, WA: Microsoft Press.
Xia, Weidong, and Gwanhoo Lee. 2004. Grasping the Complexity of IS Development Projects. *Communications of the ACM* 47 (5):68-74.

Appendix 1: Project Abandonment Points and Cumulative Resource Consumption ($)



Appendix 2: Model Equations

Work Sector Equations

Additional requirements= New Requirements
Units: tasks/Month

Apparent progress rate=Gross productivity*Workforce
Units: tasks/Month

Cumulative perceived progress=Work really done+Undiscovered rework
Units: tasks

Fraction of work made obsolete= WITH LOOKUP (XIDZ(Additional requirements,Work really done,0),([(0,0)-(100,10)],(0,0),(1,1),(1.001,1),(100,1)))
Units: 1/Month

Fraction satisfactory=0.7
Units: Dmnl

generation of rework=Apparent progress rate*(1-Fraction satisfactory)*utilization fraction
Units: tasks/Month

Known rework= INTEG (obsolescence rate+rework discovery-rework to be done,0)
Units: tasks

obsolescence rate=Work really done*Fraction of work made obsolete
Units: tasks/Month

rework discovery=(Undiscovered rework/Time to detect rework)*Work switch
Units: tasks/Month

rework to be done=Known rework/time to reintegrate
Units: tasks/Month

time to begin work=1
Units: Month

Time to detect rework=TTDRWF(Fraction perceived complete)
Units: months

time to reintegrate=1
Units: Month

Undiscovered rework= INTEG (generation of rework-rework discovery,0)
Units: tasks

utilization fraction= WITH LOOKUP (XIDZ(Work in progress,Apparent progress rate,0),
([(0,0)-(1000,10)],(0,0),(0.1,0.1),(0.5,0.5),(0.66,0.66),(0.77,0.77),(0.88,0.88),(0.99,0.99),(1,1),(1000,1)))
Units: Dmnl

work begun=Work switch*(Work to be done/time to begin work)
Units: tasks/Month

work completed=(Apparent progress rate*utilization fraction)*Fraction satisfactory
Units: tasks/Month

Work in progress= INTEG (+work begun-generation of rework-work completed,0)
Units: tasks

Work really done= INTEG (+work completed-obsolescence rate,0)
Units: tasks

Work switch= WITH LOOKUP (Project abandonment,([(0,0)-(10,10)],(0,0),(0.9,0),(1,1)))
Units: Dmnl

Work to be done= INTEG (rework to be done-work begun+Additional requirements,Initial project definition)
Units: tasks

New Requirements Generation Sector Equations

Interval for requirements addition=RANDOM UNIFORM(10, 20 , 5)

Units: months

Lower bound=150
Units: 1/Month
New Requirements=Pulse for new requirements*Range for new requirements
Units: tasks/Month

Operative project definition= INTEG (+New Requirements,Initial project definition)
Units: tasks

Pulse for new requirements=PULSE TRAIN(5, 1 , Interval for requirements addition, 360)
Units: tasks

Range for new requirements=RANDOM UNIFORM(Lower bound, Upper bound , 2)
Units: 1/Month

Upper bound=350
Units: 1/Month

Perceived Project Viability Sector Equations

adjustment time=10
Units: months

Estimated resource overruns=XIDZ((Cumulative perceived progress+(Gross productivity*Effort perceived remaining)),proj definition on books,2)
Units: Dmnl

Estimated schedule overruns=XIDZ(Scheduled completion date,Updated initial scheduled completion date,2)
Units: Dmnl

Initial project definition=5500
Units: tasks

Perception of project viability= WITH LOOKUP (SMOOTH(Resource overruns*Schedule overruns,time to perceive viability),([(0,0)-(5,1)],(0,1),(1.5,1),(1.55,0),(4,0)))
Units: Dmnl

proj definition on books=SMOOTHI(Operative project definition, adjustment time , Initial project definition)
Units: tasks

time to perceive viability=3
Units: months

Productivity Sector Equations

Cumulative perceived progress=Work really done+Undiscovered rework
Units: tasks

Effort perceived remaining=Tasks perceived remaining/Perceived productivity
Units: person*months

Fraction perceived complete=XIDZ(Cumulative perceived progress,Operative project definition,0)
Units: Dmnl

Gross productivity=1
Units: tasks/person/Month

Indicated productivity=Weight givern real productivity*Real productivity+(1-Weight givern real productivity)*Gross productivity
Units: tasks/(Month*person)

Perceived productivity= INTEG ((Indicated productivity-Perceived productivity)/Time to perceive productivity,Indicated productivity)
Units: tasks/person/Month

Real productivity=Fraction satisfactory*Gross productivity
Units: tasks/person/ Month

Time to perceive productivity=6
Units: months

Tasks perceived remaining=Operative project definition-Cumulative perceived progress
Units: tasks

Weight given real productivity=WGRPF(Fraction perceived complete)
Units: Dmnl

WGRPF([(0,0)-(10,10)],(0,0),(0.2,0.1),(0.4,0.25),(0.6,0.5),(0.8,0.9),(1,1))
Units: Dmnl

Scheduling Sector Equations

Active project= WITH LOOKUP (work completed,([(0,0)-(10000,10)],(0,0),(0.9,0),(1,1),(10000,1)))
Units: Dmnl

Completion threshold=.99
Units: Dmnl

Indicated completion date=Project time+Time perceived required
Units: Month

Net additions to schedule=(Indicated completion date-Scheduled completion date)/
Schedule adjustment time)
Units: Dmnl

Project time= INTEG (Time spent active,0)
Units: months

Schedule adjustment time=1
Units: Month

Scheduled completion date= INTEG (Net additions to schedule,initial scheduled completion date)
Units: Month

Time perceived required=XIDZ(Effort perceived remaining,Workforce sought,0)
Units: months

Time remaining= WITH LOOKUP (Scheduled completion date-Project time,([(0,0)-(600,600)],(0,0),(0.99,0),(1,1),(500,500)))
Units: months

Time spent active=Active project
Units: Dmnl

Project Re-evaluation and Post-hoc Reassessment Sector Equations

Increase in date=((1-Project abandonment)*initial scheduled completion date)/Time to update schedule
Units: Dmnl

initial scheduled completion date=80
Units: Month

New iteration scheduled completion date= INTEG (Increase in date,initial scheduled completion date)
Units: months

Project abandonment= WITH LOOKUP (Perception of project viability,([(0,0)-(1,1)],(0,0),(0.01,0.01),(0.05,0.02),(0.09,0.03),(0.095,0.05),(0.1,1),(0.2,1),(0.25,1),(0.3,1),(1,1)))
Units: Dmnl

Time to update schedule=2.5
Units: Month

Workforce Sector Equations

hiring rate=Restrictions on ability to adjust workforce*(((Workforce sought)/WorkForce adjustment Time)*Perception of project viability)
Units: persons/Month

Indicated workforce=XIDZ(Effort perceived remaining,Time remaining,0)
Units: persons

initial workforce estimate=70
Units: people

Project abandonment= WITH LOOKUP (Perception of project viability,([(0,0)-(1,1)],(0,0),(0.01,0.01),(0.05,0.02),(0.09,0.03),(0.095,0.05),(0.1,1),(0.2,1),(0.25,1),(0.3,1),(1,1)))
Units: Dmnl

reassignment rate=((Workforce*(1-Project abandonment))/time to reassign)
Units: people/Month

Restrictions on ability to adjust workforce= WITH LOOKUP (XIDZ(Workforce,initial workforce estimate,2),([(0,0)-(2,1)],(1,1),(1.00917,0.635965),(1.0581,0.298246),(1.12538,0.114035),(1.33333,0.0263158),(2,0)))
Units: Dmnl

Time remaining= WITH LOOKUP (Scheduled completion date-Project time,([(0,0)-(600,600)],(0,0),(0.99,0),(1,1),(500,500)))
Units: months

time to reassign=0.25
Units: months

WCWFF([(0,0)-(40,1)],(0,0),(3,0),(6,0),(9,0.1),(12,0.3),(15,0.7),(18,0.9),(21,1),(40,1))
Units: Dmnl

Willingness to change WorkForce=(WCWFF(Time remaining))
Units: Dmnl

Workforce= INTEG (hiring rate-reassignment rate,initial workforce)
Units: people

WorkForce adjustment Time=6
Units: months

Workforce sought=Willingness to change WorkForce*Indicated workforce+(1-Willingness to change WorkForce)*Workforce
Units: persons