Applying System Dynamics to the Product Development Process

by

Nancy Riley Jankowiak

B.S., Materials Science and Engineering
Massachusetts Institute of Technology, 1993

Submitted to the System Design and Management Program
in Partial Fulfillment of the Requirements for the Degree of

Master of Science in Engineering and Management

at the

Massachusetts Institute of Technology
February, 2000

Signature of Author.......................
System Design and Management Program
January 11, 2000

Certified by...............................
James H. Hines
Senior Lecturer, Sloan School of Management
Thesis Supervisor

Accepted by...
Thomas E. Kochan
George Maverick Bunker Professor of Management
Co-Director, System Design and Management Program

Accepted by............................
Paul A. Lagace
Professor of Aeronautics & Astronautics and Engineering Systems
Co-Director, System Design and Management Program

Applying System Dynamics to the Product Development Process

by

Nancy Riley Jankowiak

Submitted to the System Design and Management Program
on January 11, 2000 in Partial Fulfillment of the Requirements for the Degree of
Master of Science in Engineering and Management

ABSTRACT

A system dynamics study of the product development process at a given company was conducted. The objective was to determine whether system dynamics could be used to understand and improve the process of product development in this particular setting. A team of experts representing the key departments within the company was assembled and used as a resource throughout the project. A standard system dynamics method was used.

Profitability was identified as the key parameter on which the study should focus. Behavior patterns in the product development process were identified by the team, and hypotheses were developed using causal loop diagramming. System dynamics models were then created using Vensim software. These models were simulated and analyzed to provide insights into the real world system which exists at the company.

Key findings included the identification of the tension between management goals and competitive pressures, the potential side effects of cutting costs by reducing resources, and the importance of productivity and the cost of resources. These insights enhanced the company's understanding of their current process and helped them to develop a more systemic view of their problems. Further work will be required to allow full policies for improving profitability to be formulated and tested. The conclusion of this thesis is that system dynamics can and has helped this company to understand and improve its product development process, and the benefits can be enhanced through further application of these techniques.

Thesis Supervisor: James H. Hines
Title: Senior Lecturer, Sloan School of Management

Acknowledgments

Many people helped to make my pursuit of a masters degree, and my completion of this thesis, possible. I am extremely grateful to all of them.

Special thanks go to Jim Hines for introducing me to system dynamics in his classes, and for guiding this thesis as my advisor. His engaging teachings sparked an interest in system dynamics in me, as in many of my fellow students, that I hope to continue to pursue. Appreciation also goes to his great teaching assistants: Jody House, Leslie Martin, Mary Tolikas, and Paulo Goncalves. I would also like to thank Jim Lyneis, who provided his valuable perspective on system dynamics and its applications in several courses.

Many thanks are owed to the staff of ProductCo. The team of experts which provided the main body of knowledge for this thesis, as well as the many others who gave their thoughts on particular areas, were wonderful resources. I am grateful for their time and patience in working with me on this project.

Thank you to my company for sponsoring my participation in the System Design and Management Program. Thanks also to the staff of the SDM Program for their help in negotiating my way through it all.

I am deeply indebted to my classmates in the SDM Program. Their varied perspectives made this a rich and unique learning experience. Their knowledge, friendship, and support have been invaluable.

Most of all, I thank my family. My parents taught me early on that education is the greatest gift that you can give to yourself. I have been inspired by them and by my sisters throughout my life. I am profoundly grateful to my husband Rob for his support of my participation in SDM. Without him, I would never have been able to do it.

Table of Contents

Introduction: Profitability and Product Development

In a complex and competitive market, companies that design and manufacture products have many issues on which to focus. Quality, time to market, customer needs, new technologies -- these things and many others must be considered in running a successful enterprise. However, in the end what keeps the cycle of designing and releasing new products going is money.

It is often difficult for leaders, especially in large companies, to see what impact their policies and decisions will have on the bottom line. This can be extremely complicated in the product development process, where a decision made today may only show its full ramifications several years in the future. At the company studied in this thesis, management was concerned that their level of profitability was not competitive. They suspected that some of the policies and processes embedded in their product development cycle were hindering their efforts to improve profits.

Given the complex nature of the product development process and the difficulty of foreseeing the consequences of many actions, system dynamics appears to be a good tool to apply to this problem. The purpose of this thesis was to determine if system dynamics would be useful in helping managers at this company to see how their policies impact profitability. Further, we hoped to identify leverage points for future policies that would help this company reach its desired profit level.

Research Method

In this project I studied a company, fictionalized here as "ProductCo", which designs and manufactures components that are supplied to other companies. These components are incorporated into the other companies' consumer products.

There are several characteristics of ProductCo which are important to this thesis. The company's components, or "projects", are developed over several years and involve a large number of tasks which are performed mainly by engineers. ProductCo bids on and develops projects for a variety of consumer product manufacturers within its industry. The level of competition is extremely high overall, although it varies over the various types of products which ProductCo supplies.

All of the research for this thesis was conducted on site at ProductCo. The primary method used was interviews, both one-on-one and in larger group sessions. Quantitative data were also obtained from ProductCo.

A "client" in the upper management of ProductCo was identified at the outset, as well as a small core team which participated throughout the project. The team included representatives from engineering, engineering management, business planning, and manufacturing. This covers the major departments which are involved in developing and building products, as well as estimating costs and determining pricing.

The Current State of Affairs

ProductCo has an established product development process, as well as a system for bidding and accounting for costs of projects. This section gives an overview of these as they existed at the start of this thesis project.

The process begins when a customer comes to ProductCo for a bid on a project to design and supply a particular component. This Request for Quote (RFQ) includes a description of the desired component and a list of the specifications that it must meet, as well as an indication of any governmental regulations that will affect the component. The Marketing and Sales department is responsible for putting together the quote response. They must hold a business case review with top management to ensure that it makes strategic sense for ProductCo to go after this project. Based on the business case and the customer who is requesting the quote, the project is assigned to Category A or Category B.

A Category A project is one that ProductCo views as a "must-win" contract. This may be with an established customer, or with a new customer with which the company strongly desires business. It may also be a project that would fill a particular opening at one of ProductCo's existing facilities. These are pursued more aggressively than Category B projects, which ProductCo would like to win but are not viewed as being as essential to the company's future.

Using the category designation as well as historical information from past projects, Marketing and Sales develops a quote which is submitted to the customer. The quote package includes dollar figures for investment and for price per part once the component is in production (piece price). It also includes a

Statement of Work than indicates specifically what design and engineering services will be provided by ProductCo, and what will be the responsibility of the customer.

There may be some negotiation, and then the customer makes their decision. If the quote is accepted, the resources needed for the project are identified and the team is formed. At this stage, key subsuppliers, if any, are identified and sourced. The supplier engineers then become members of the project team.

The process used for the actual design of the component varies widely. A component that includes software, for example, has a much different design procedure than a component that is strictly mechanical. Also, some customers source ProductCo with an entire subsystem, which requires more coordination and control activities than a smaller project might. The customer's own product development process also has an effect; different customers have different requirements for design reviews, prototype builds, and other deliverables.

The design process can be generalized as a set of tasks that must be performed in order for ProductCo to progress from a clean sheet of paper to a completed and validated product design. A task is defined as the unit of work in this process. Sets of these tasks may add up to activities which include creating drawings using CAD (computer aided design), building prototypes, running CAE (computer aided engineering) simulations, writing software, researching competitive products, and performing testing, among other activities.

It is important to note that during the design process, changes to the initial requirements are often made. This could be due to a request from the customer, or due to factors discovered during the design process that were not previously known. Depending on the situation, the quoted investment or piece price may be renegotiated.

Once the design is complete and is approved by the customer, ProductCo orders or fabricates any special tooling or facilities that are required to produce the component. This phase may overlap the design phase, where possible, in order to reduce the total time required.

The production parts must be validated through testing and in reviews with the customer. Changes are made to the production tools or processes, if required, in order to a provide parts that meet the customer's specifications and any relevant regulations. Once the production components are fully approved, the ramp-up to full production can begin. Support is provided at the customer's facility as needed for the first builds of the customer's product.

The Problem

ProductCo is in an extremely competitive industry. Many of its smaller rivals have gone out of business, or have been involved in mergers and takeovers. Until recently, ProductCo had been essentially a captive supplier to a single customer. As it tries to branch out and broaden its customer base, ProductCo feels that it must improve its profit margins in order to survive in this competitive environment.

Figure One shows this concern graphically. Profitability, as measured by Return on Sales (ROS), has hovered at a relatively steady level in recent years. ProductCo fears that this level is not high enough to sustain the company; in five years, the company may well be forced to merge or go out of business.

They hope that this can be avoided by finding a way to reach a level of ROS that is comparable to its most successful competitors.

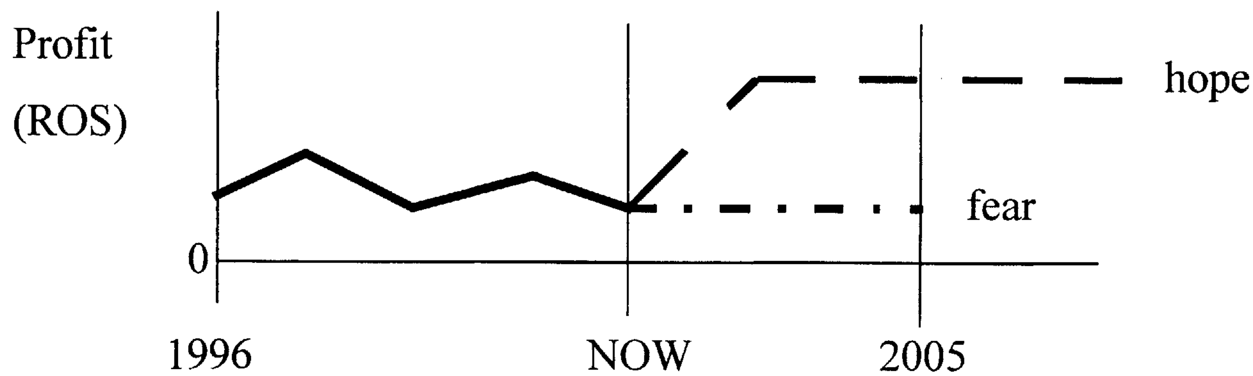


Figure 1: The Problem With Profitability

There are many ways to increase profit. The most obvious is to increase price, though in a competitive environment that may not be advisable. Cost cutting measures may be taken, but they may have unwanted side effects. To help ProductCo evaluate these and other potential policies, I undertook a system dynamics analysis of the problem.

The Basics of System Dynamics

System dynamics is "a powerful method to gain useful insight into situations of dynamic complexity and policy resistance."[1] That is, we can use it as a tool to help us understand what is going on in the

[1] Sterman, J. D. 1998. *Business Dynamics: Systems Thinking and Modeling for a Complex World.* Partial Draft, Version 1.0, May 1998. Page 1-49.

product development process that is affecting profitability, and why actions that we take to improve our situation don't always work as expected.

System dynamics represents the real world system using causal loop diagrams and explicit models with stocks, flows, and information feedback structures. The basic concepts of these tools are presented in this section to aid the reader in understanding the analysis which follows.

Causal loop diagrams are used to represent the linkages between the factors, or variables, in the system. They are often used to show a person or team's dynamic hypothesis - their opinion, based on experience, of how a change in one variable drives changes in others. The underlying assumption is that there is feedback in all systems - that is, our actions today influence the situation that will exist in the future. Thus we have causal *loops*, not causal *lines*. Figure Two shows a simple example of a loop created by our decisions impacting our environment.

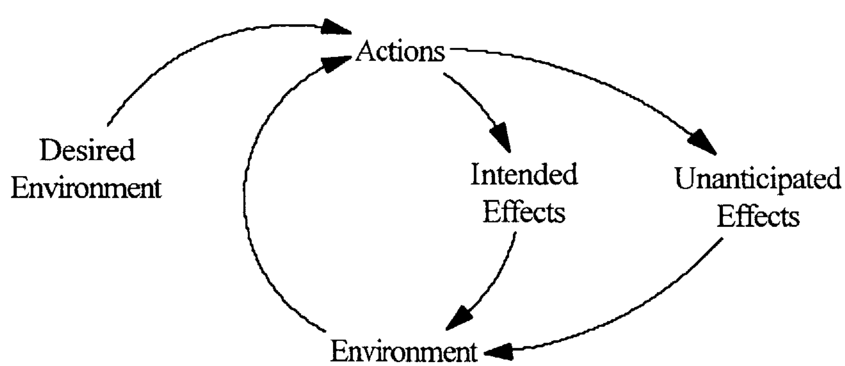


Figure 2: Simple Causal Loops

In this loop, we take actions based on the current state of our environment as well as the desired state. These actions have effects, which may be as planned or unintentional, that change the state of the environment and influence our future actions.

The links between variables in a causal loop diagram of a specific system are positive or negative in sign. If, as variable A increases, it causes variable B to increase, there is a positive link from A to B. If, on the other hand, variable B decreases as A increases, there is a negative link from A to B. Examples of these linkages can be observed in Figure Three.

Each loop also has a sign. Positive loops tend to reinforce or amplify whatever is happening in the system, while negative loops counteract and oppose change, seeking to restore balance.[2] An example of each is shown in Figure Three. The sign of a loop can be determined by starting with a change in one variable, tracing the behavior of the variables around the loop, and observing whether it tends to reinforce the change in the first variable or counteract it. All dynamics in a system are caused by these positive or negative loops, or combinations thereof.

---

[2] Sterman 1998, page 1-13.

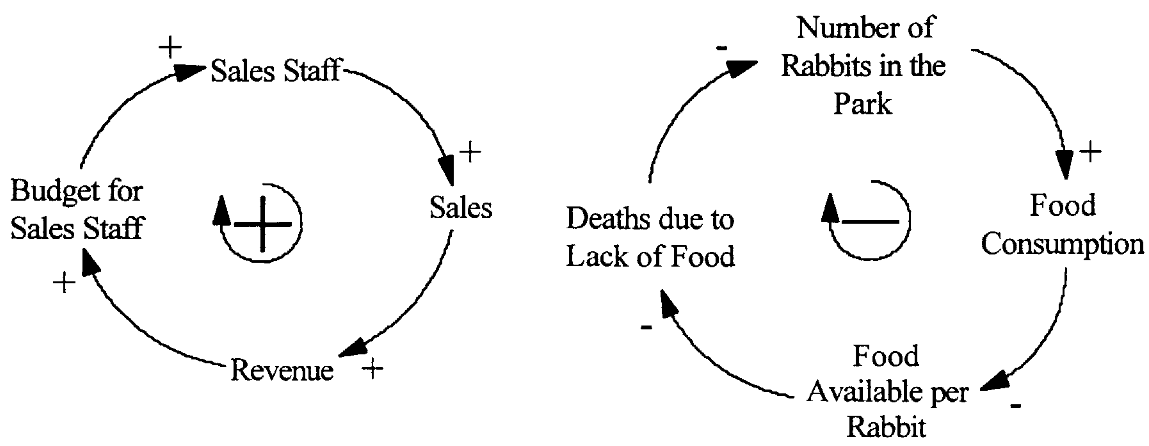


Figure 3: Positive and Negative Loops

The system can also be represented in a model, using stocks, flows, and information feedback. This is often done after dynamic hypotheses are proposed and diagrammed in causal loops. A simple stock and flow diagram is shown in Figure Four.

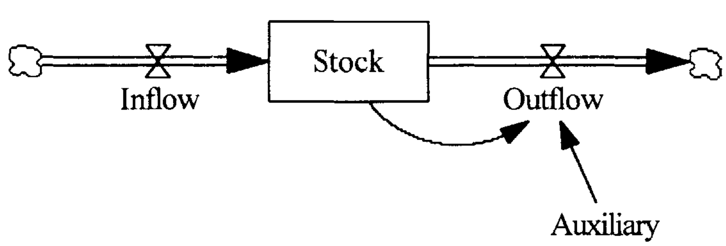


Figure 4: Stock and Flow Diagram

Boxes indicate stocks, double arrows with valves indicate rates of flow, single arrows represent information links, and clouds show the boundaries of the model. A variable shown in sharp brackets, such as <variable>, is a shadow variable. This means that it is shown elsewhere in the model; the "shadow" copy allows it to connect back while keeping the diagram neater and more readable.

Each variable has units; for example, a stock might have units of dollars, tasks, items, or some other measurable, while a flow would generally have units of the measurable over time, such as tasks per month.

Underlying the diagram are equations which explicitly state the relationship between each variable. A stock is an integration of all of the rates of flow into or out of it. In the diagram in Figure Four:

$$Stock_T = Stock_0 + \int_0^T (\inf low_t - outflow_t)dt$$

An information link means that the indicated variables are included in the equation for the variable to which they point. In the diagram in Figure Four, one possibility would be:

$$\text{Outflow} = \text{Stock} / \text{Auxiliary}$$

This would indicate that the outflow is dependent on the current level of the stock, and some auxiliary which might be the time required to drain the stock. It is important to note that the units on both sides of each equation must agree. This helps to ensure that the equation is realistic in terms of the real world meaning of the variables.

In this thesis, the models, which include the stock and flow diagrams and the associated equations, were built using Vensim[3] software. This allows the model to be simulated over a given time period. The behavior of the variables can then be shown and analyzed in order to better understand why the model does or does not behave the way we suppose that it will, or the way that the real world system does.

---

[3] Vensim is a product of Ventana Systems, Inc. See http://www.vensim.com for more information.

Analysis of ProductCo's Problem

How can system dynamics be applied to help us understand ProductCo's issues with profitability? In order to answer this question, I followed a series of steps with the client and team at ProductCo. Variables and their associated reference modes were identified, dynamic hypotheses were proposed and expressed in causal loop form, and these hypotheses were modeled, expanded, and analyzed. Finally, some conclusions and plans for future work were developed.

Variables and Reference Modes

The first step in this process was to develop a list of the factors in the product development system that could be influencing the success and failure of ProductCo's efforts. An extended kickoff meeting was held with the team to get the project off to a running start. At this point, profitability had not yet been identified as the main problem to be addressed. Therefore, a wide variety of variables were proposed, and there was much discussion around what the "real" problems in the system were.

Through interviews with the client and team members, the focus was narrowed to the main problem of profitability. This was the first insight, and though it appears obvious, it was not the first thing that came to the team members' minds. The list of variables was revisited, and those that were deemed relevant to the problem were selected. Reference modes were developed for certain key variables.

These variables were put to use in creating the dynamic hypotheses that the team felt were representative of the dynamics in the real world system. Two main hypotheses were studied in this thesis. The first, "Pressure to Increase Margins", says that profit margins will adjust over time to reach the goal set by management. The second, "Cost Cutting Through Resource Reduction", indicates that if costs are too high, they will be reduced first by reducing resource levels, which may have unintended side effects. The causal loop diagrams for these hypotheses are shown in Figure Five. They will be explained, piece by piece, in the following sections.

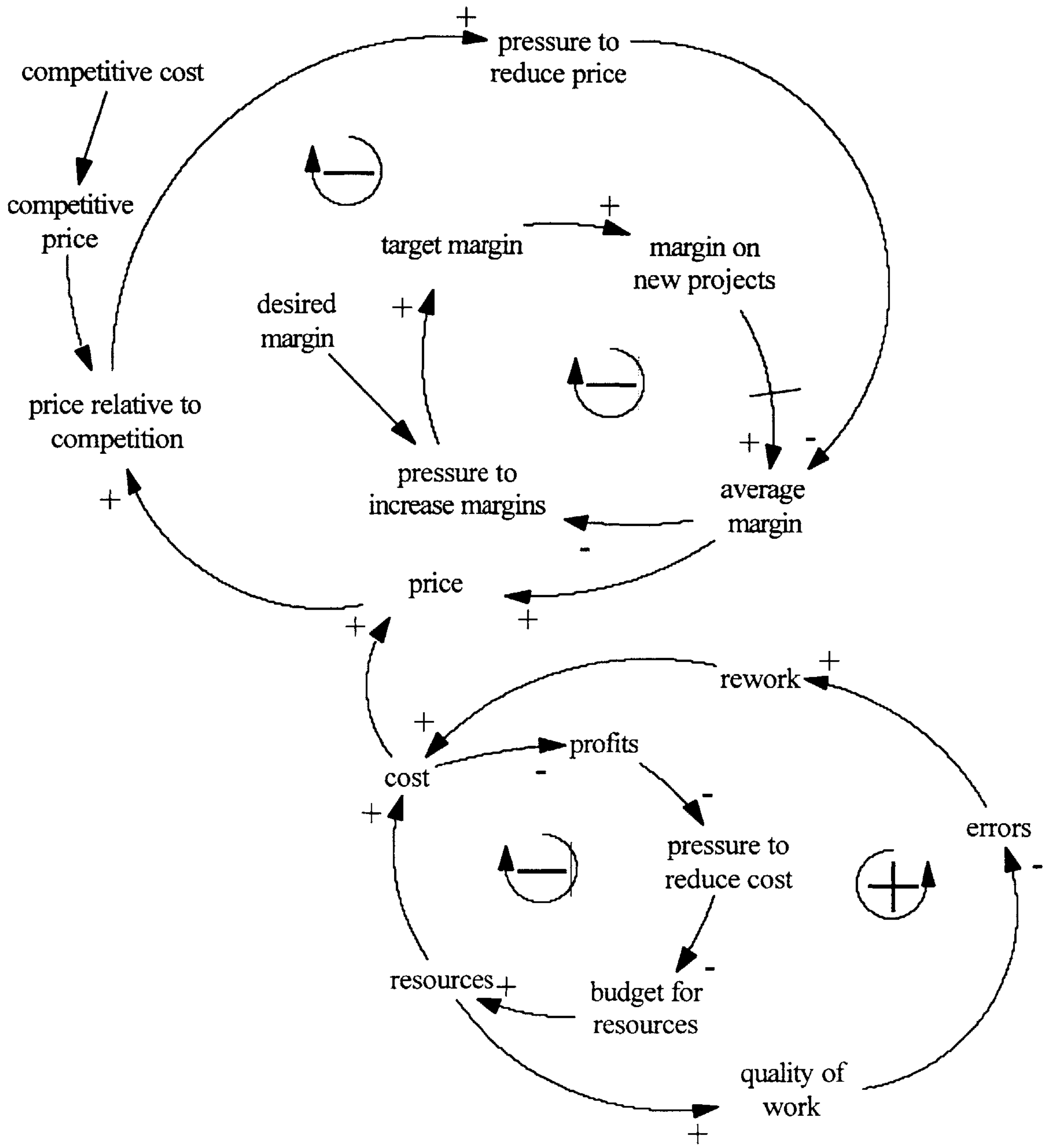


Figure 5: Causal Loops

Clearly these two hypotheses and their associated causal loops do not describe all of the dynamics operating in the product development process at ProductCo. One of the objectives of this thesis, therefore, is to determine whether a non-comprehensive set of hypotheses can still yield quality insights into the problem at hand.

Hypothesis One: Pressure to Increase Margins

One of the most basic dynamics impacting the profit level at ProductCo is the pressure to reach the desired degree of profitability. The profit which management desires, expressed as return on sales (ROS), has been identified as a particular percentage. When Marketing and Sales is responding to a request for quote (RFQ), they take this desire into consideration when developing a price. The hypothesis here is that if the desired margin is increased, this will create pressure on the system, specifically on Marketing and Sales, to increase bid margins, which over time will bring the average margin to the desired level. The causal loop diagram for this hypothesis is shown in Figure Six. The impact of competition, which limits the price that customers will be willing to pay, is added later in this section.

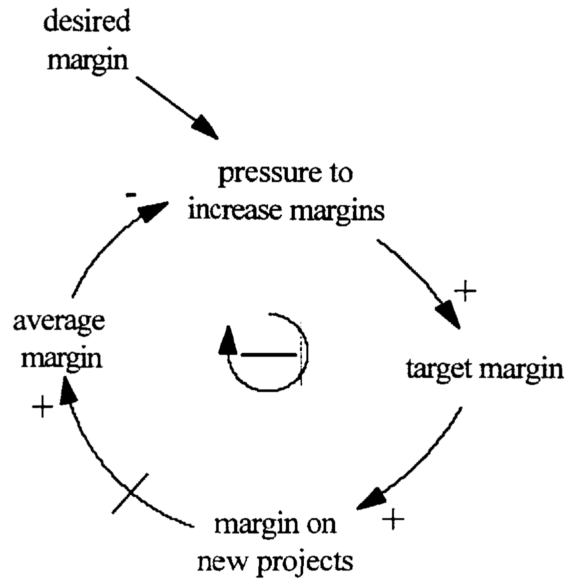


Figure 6: Causal Loop Diagram for "Pressure to Increase Margins"

Note that this is a negative loop; it tends to bring the change in the margin to a stop once the desired margin is reached. The strike through the link from "margin on new projects" to "average margin" indicates that there is a delay from the time that new projects begin to be won with higher margins and the time that the average margin actually increases.

This hypothesis can be tested by modeling the system using Vensim and running a series of simulations. Pieces of the model and its output will be presented here; the full model documentation, including all equations, can be found in Appendix A.

The basic hypothesis is complicated somewhat by ProductCo's differentiation between A and B projects. There are different target ROS percentages applied to the different categories. This is because A projects are to be pursued more aggressively, which may require a smaller profit in order to win them. B projects will have a higher margin since they are not as critical to the business and will be quoted less aggressively.

The first piece of the model shows the flow of projects. It represents projects becoming available by RFQ's being issued and ProductCo responding with bids, and the projects being won or lost. Projects which are won become active - this means that they are ongoing and are bringing in money. Eventually, they obsolete out of the system when they are completed. The stock and flow diagram for this portion of the model is shown in Figure Seven.

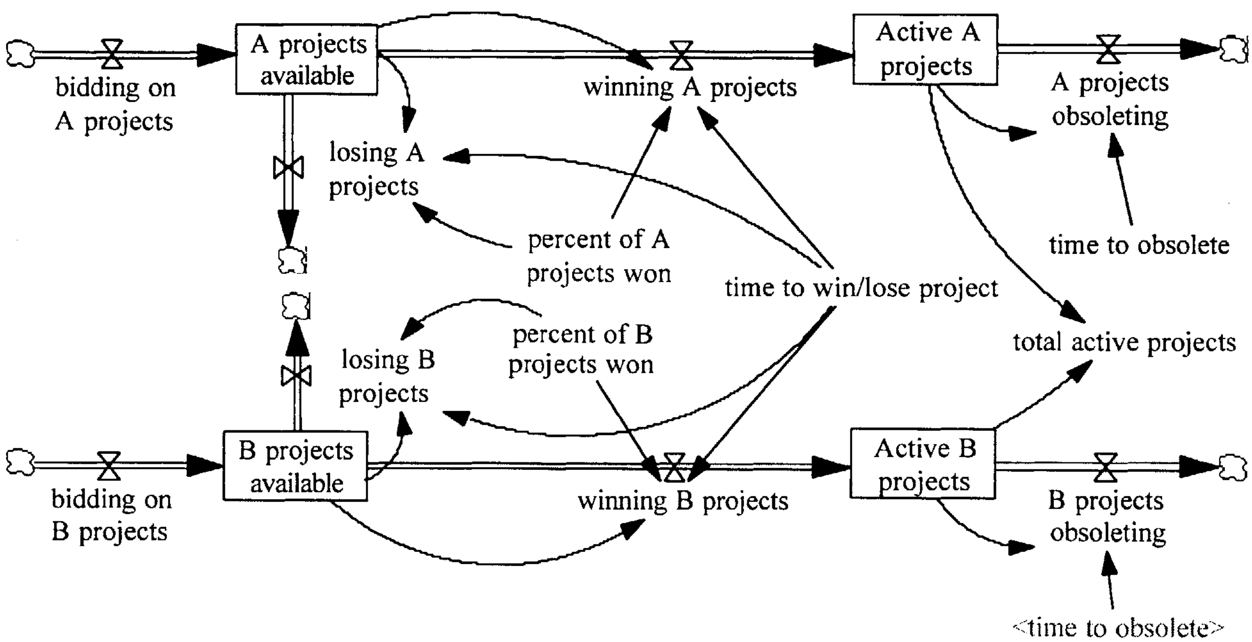


Figure 7: Model for "Pressure to Increase Margins", Part One

Some of the key equations in this section of the model are listed below. Note that those for B projects are the same as those for A projects.

The number of projects available is determined by integrating the number bid minus those that leave this stock by being won or lost:

A projects available= INTEG (bidding on A projects - (winning A projects+losing A projects))
Units = projects

The rate of winning is determined by the win percentage and the time it takes for a decision to be made by the customer:

winning A projects=(A projects available*percent of A projects won)/"time to win/lose project"
Units = projects/month

The number of active projects is the integral of those that are won minus those that obsolete:

Active A projects= INTEG (winning A projects-A projects obsoleting)
Units = projects

Associated with each of the active projects is a profit. The second piece of the model shows how the profit margin is determined and how it is affected by the desired margin. The stock and flow diagram for the second portion of the model is given in Figure Eight.

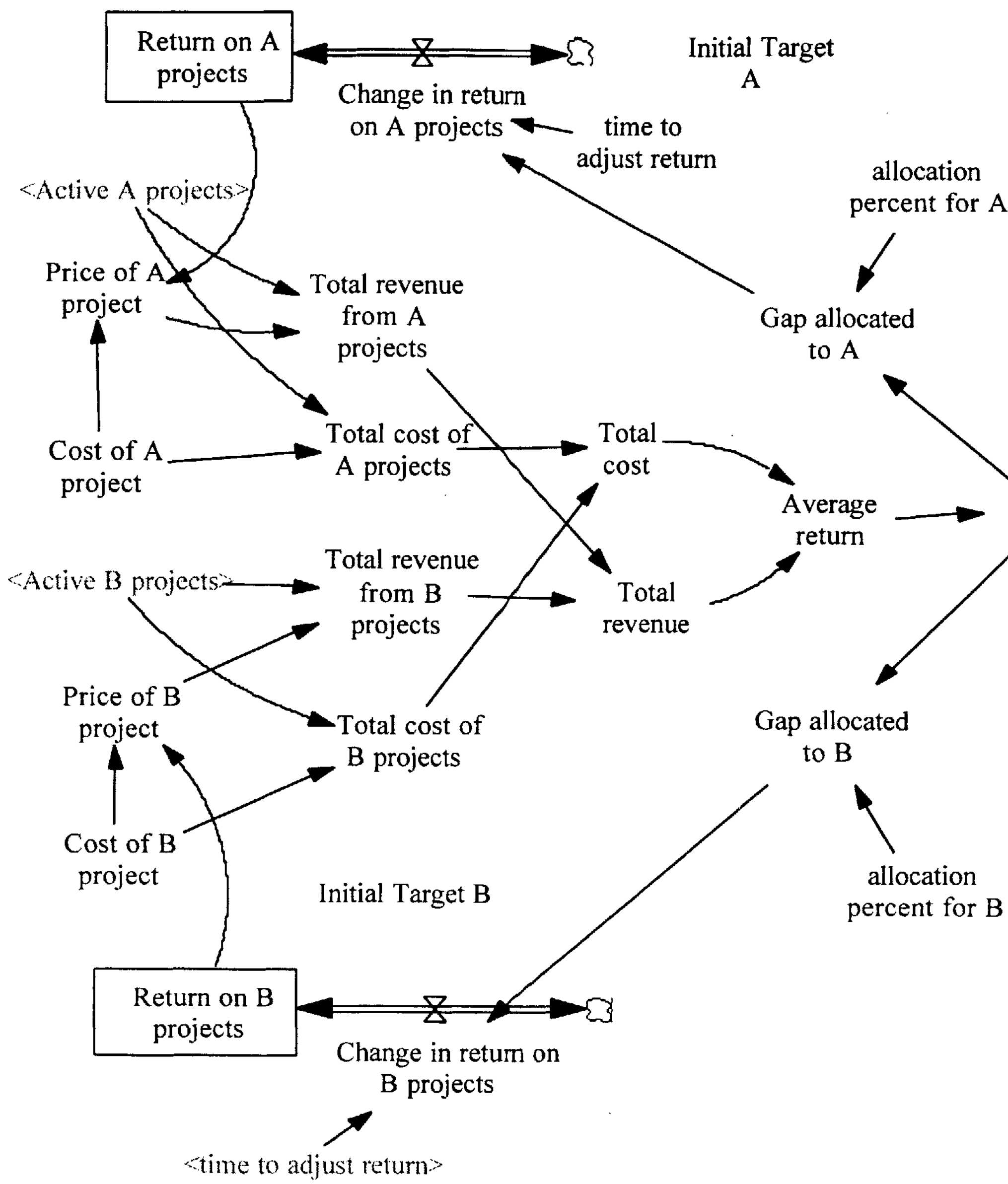


Figure 8: Model for "Pressure to Increase Margins", Part Two

This section of the model has many equations and flows of information. The major equations are listed below. Again, those for B projects are the same as those for A projects.

Price of a project is determined by its estimated cost and the return percentage currently being applied:

Price of A project=Cost of A project*(1+Return on A projects)

Units = dollars/project

The average return is calculated by considering the price and cost of all the active projects:

Average return=(Total revenue/Total cost)-1
Units = fraction

The return to be applied to projects is determined by the gap between the average return and the desired return[4]. A portion of this gap is allocated to A projects, a portion to B projects:

Return on A projects= INTEG (Change in return on A projects)
Units = fraction

Change in return on A projects=Gap A/time to adjust return
Units = fraction/month

Gap allocated to A=Gap in average*allocation percent for A
Gap in average=Desired average return-Average return
Units = fraction

Given this model, one can predict that it will behave as the hypothesis indicates. That is, the average margin will increase until it meets the desired margin. Indeed, when the simulation is run, this is what happens. Figure Nine shows the behavior of the average margin over time. The desired margin in this case was set to 8%.

---

[4] Note that there is no first order control to prevent this stock from going negative. This is not anticipated to be an issue within the range of values in this system. A negative stock could occur if the desired return is negative, or with extreme values for allocation factors and a large difference in returns on A and B projects.

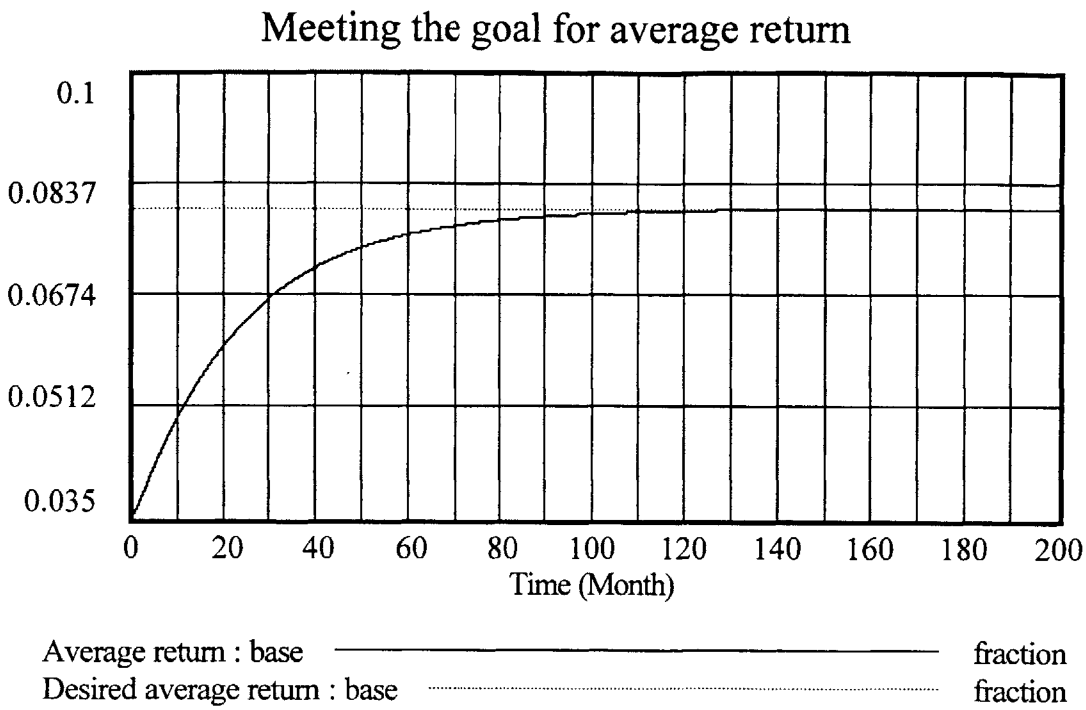


Figure 9

In order to attain this goal, the margins on A and B projects, and by extension their prices, must have increased over time as well. This is shown in Figures Ten and Eleven.

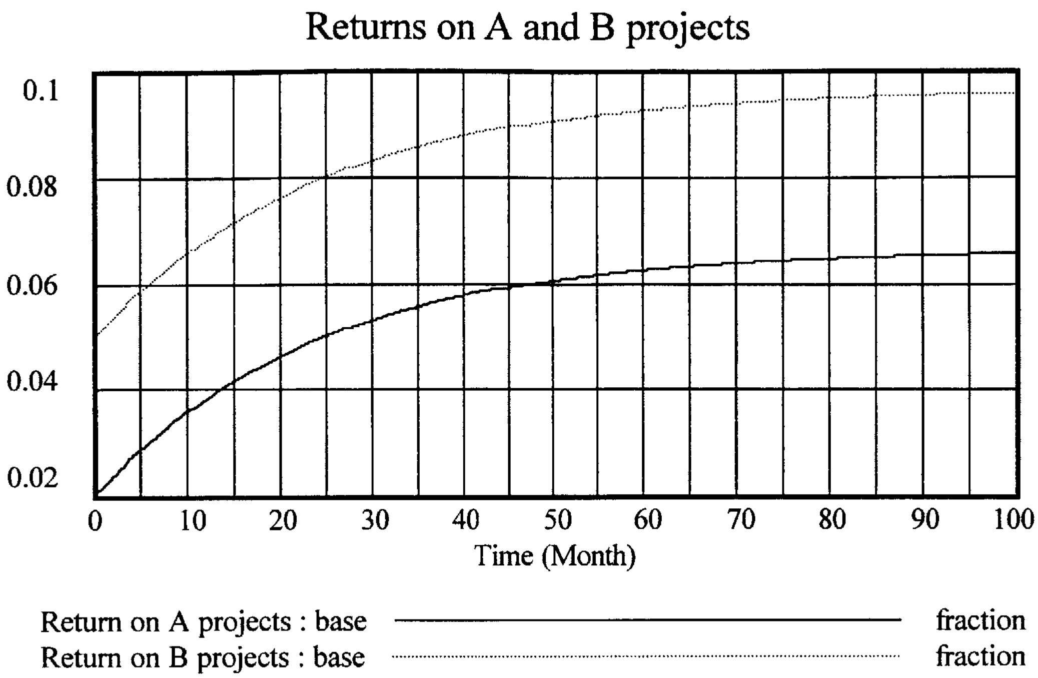


Figure 10

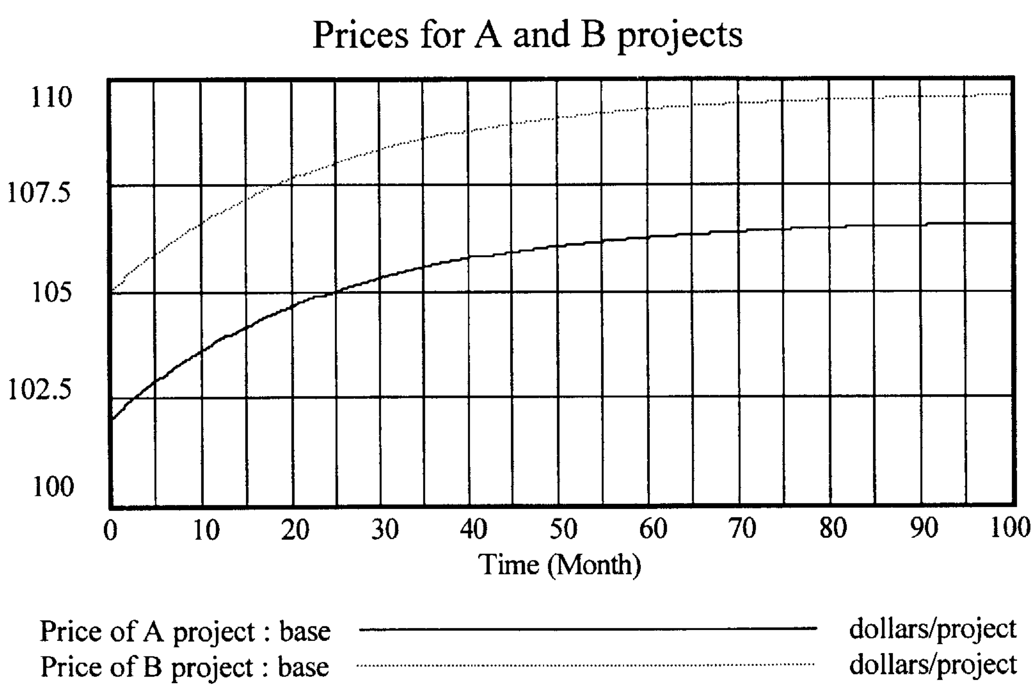


Figure 11

This output raises the issue of competition; clearly price cannot be allowed to rise unchecked to meet management's desires for ROS. There must be a second negative loop which counteracts this growth. In ProductCo's case, this limiting factor is the price that the customer is willing to pay. This, as mentioned, stems from competition.

In order to capture this important fact, the causal loop diagram was modified, and additional variables and equations were added to the model. The new causal loop diagram is shown in Figure Twelve.

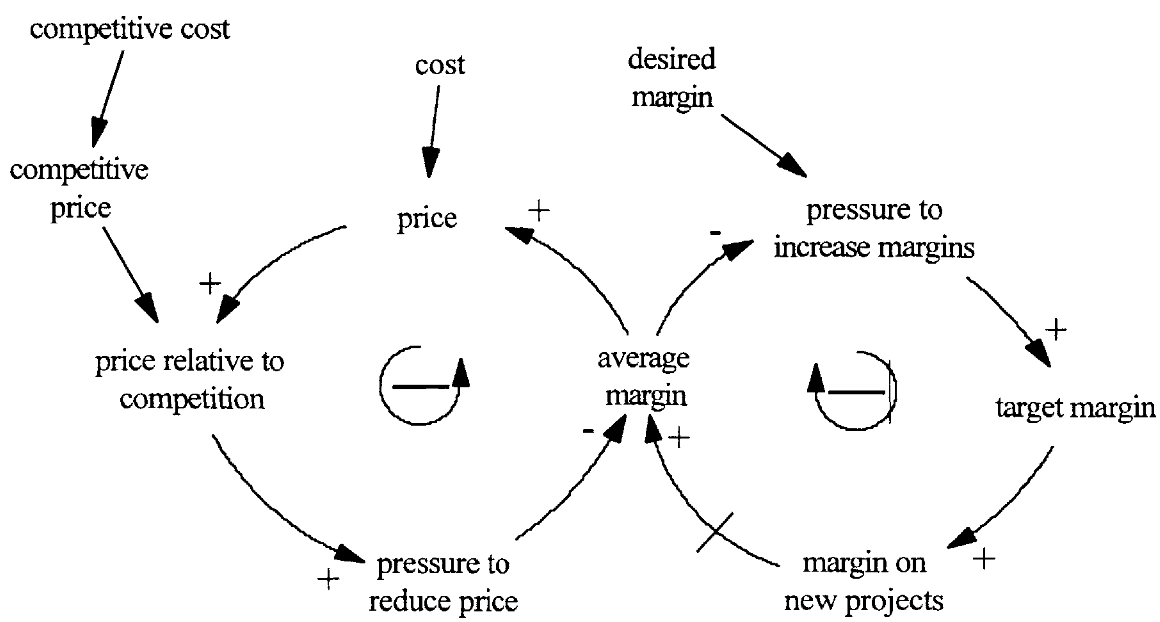


Figure 12: Modified Causal Loop Diagram for "Pressure to Increase Margins"

Structure was added to the model which allows ProductCo's price relative to that of the competition to affect the margin which is actually applied when the price is determined. The changes to the model for A projects are shown in Figure Thirteen; only the modified portion is given here. Note that "Return on A projects" has been changed to "Indicated Return on A projects"; this is because now the return which is indicated by management's desires may not be what is actually applied for pricing due to competitive factors. The same changes were made to the structure which applies to B projects.

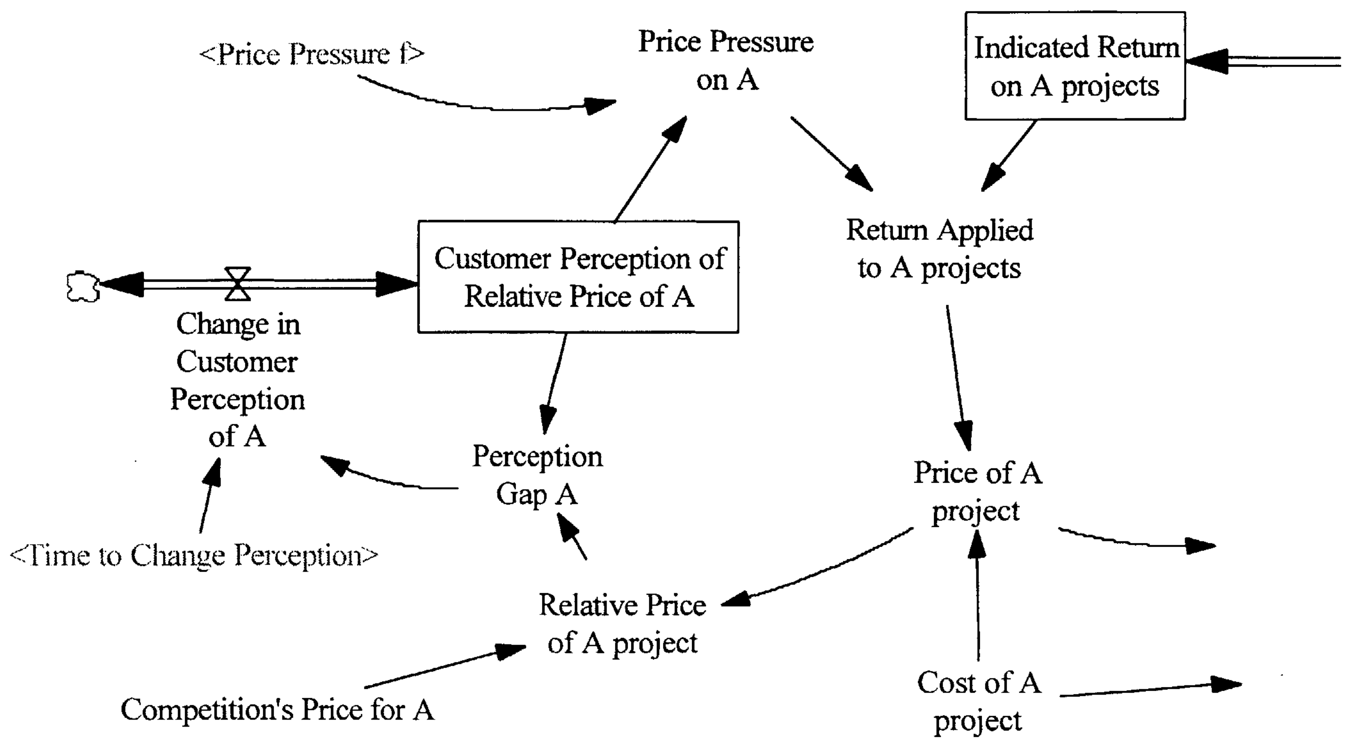


Figure 13: Modified Portion of Model for "Pressure to Increase Margins"

A delay occurs between the relative price changing and the customer perceiving this change. This is due to the fact that customers do not review prices on an instantaneous basis; realistically, they would do this periodically. In this case, a quarterly review is assumed. Other key equations are given below.

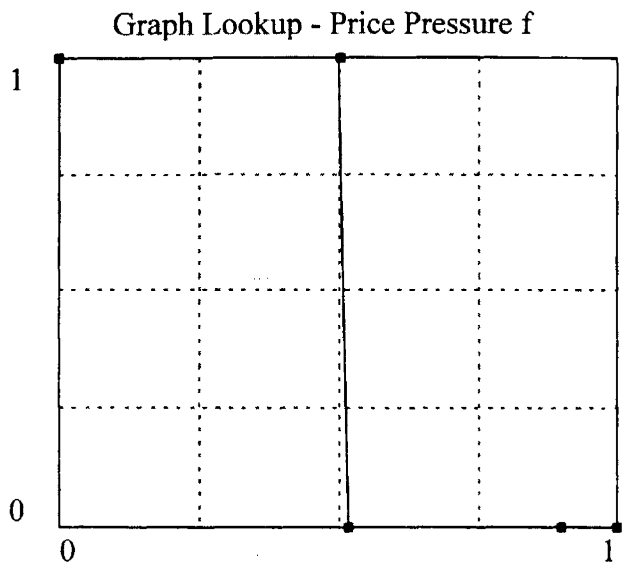
Price pressure is applied by adding an effect factor to the margin that is actually used to calculate price:

Return Applied to B Projects=Price Pressure on B*Indicated Return on B projects
Units = fraction

Price pressure is a table function with the customer's perceived relative price as an input:

Price Pressure on A=Price Pressure f(Customer Perception of Relative Price of A)

Units = dimensionless



This table function effectively allows margins to be set at the indicated level if ProductCo's price is below the competition's, while dropping quickly to nearly zero when it is above the competitor's price.

When the model is simulated with the effects of price pressure added, it shows that the desired margin cannot be reached. Figure Fourteen shows this behavior.

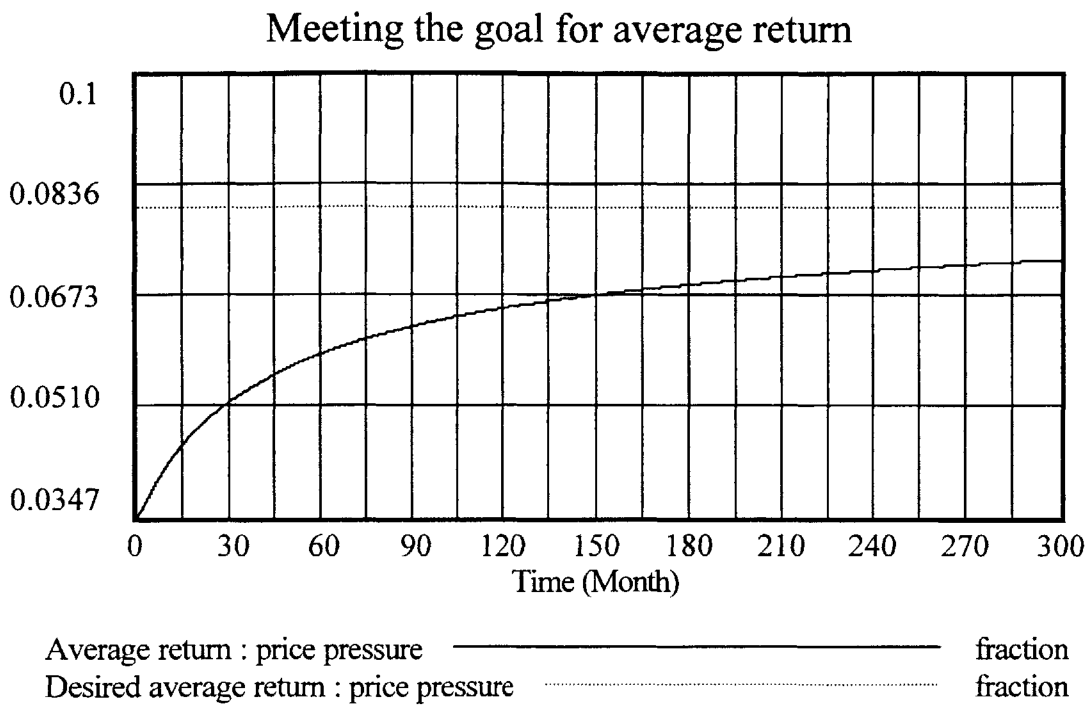


Figure 14

After further analysis, it is shown that this is only true if the competitor's price is above ProductCo's cost to execute the project, but below the price that includes the margin that is indicated to reach the desired average margin. As is shown in Figure Fifteen, if the competitor's price is below ProductCo's cost, almost no margin can be charged. This would likely drive ProductCo out of business, or at least away from the projects of this kind. If the competitor's price is high, ProductCo can charge the prices that they need to reach their desired margin, and the goal of 8% is achieved.

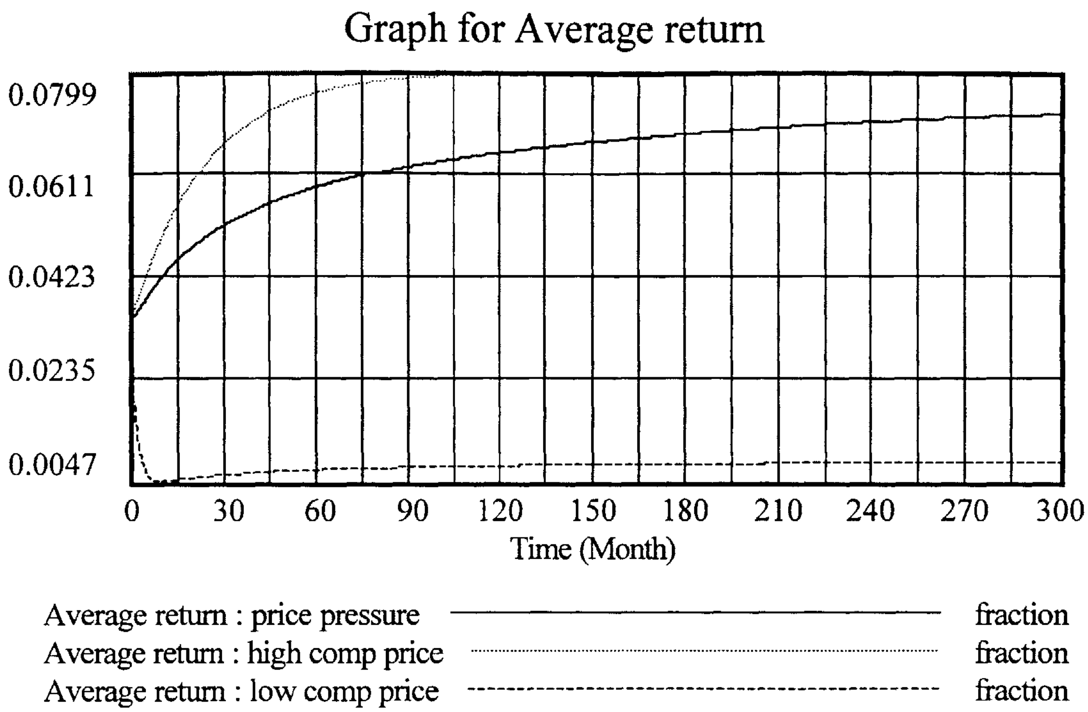


Figure 15

Another fact which is interesting to note is that management's goal continues to rise despite the competitive pressures that are keeping the actual margins down. This is shown in Figure Sixteen.

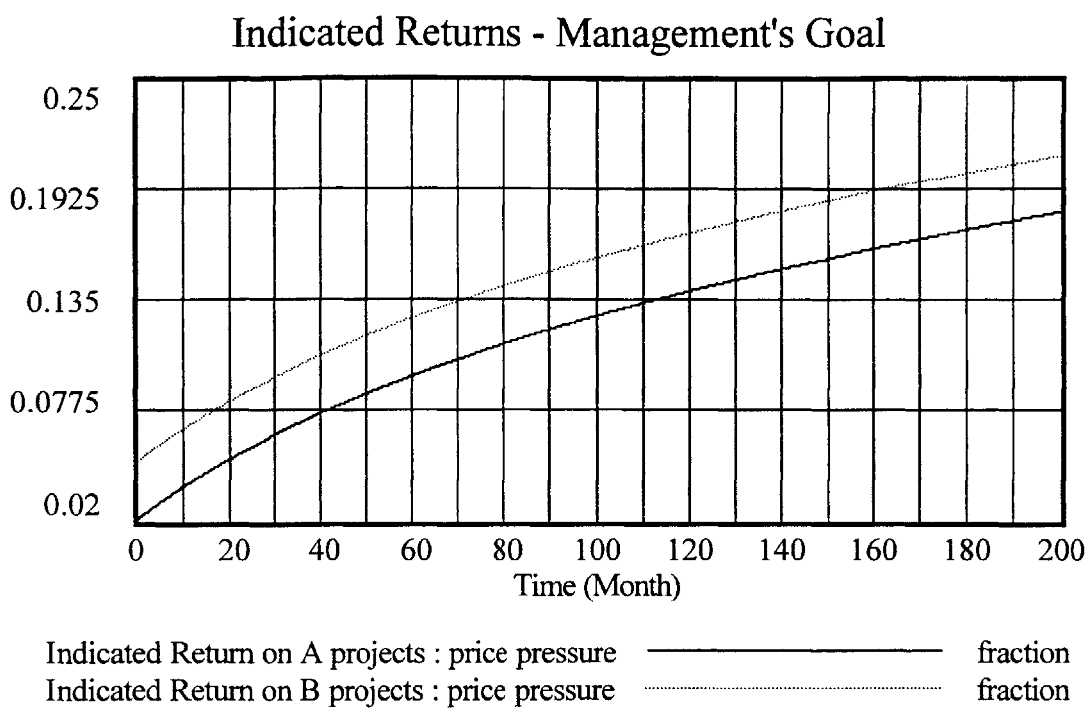


Figure 16

What does this mean in the real system? It is likely that management will see what is going on and will not continue to have unrealistic goals. However, it is indicative of the ongoing conflict between desired margin and competitive pressures within the company. ProductCo must then seek ways to relieve this situation.

Given that there is price pressure from the competition that might prevent ProductCo from reaching its goal, what can be done? One avenue to pursue is cost reduction. The next hypothesis proposed addresses this issue.

Hypothesis Two: Cost Cutting Through Resource Reduction

When there is pressure in the system to cut costs, a common response is to reduce resources. This is often done through workforce reductions, either by layoffs or attrition. Hypothesis two maintains that while this reduces costs in the short term, there are often unintended side effects. One potential effect is that the quality of work done is reduced, since the company is trying to do the same amount of work with fewer resources. Figure Seventeen shows the causal loop diagram for Hypothesis Two.

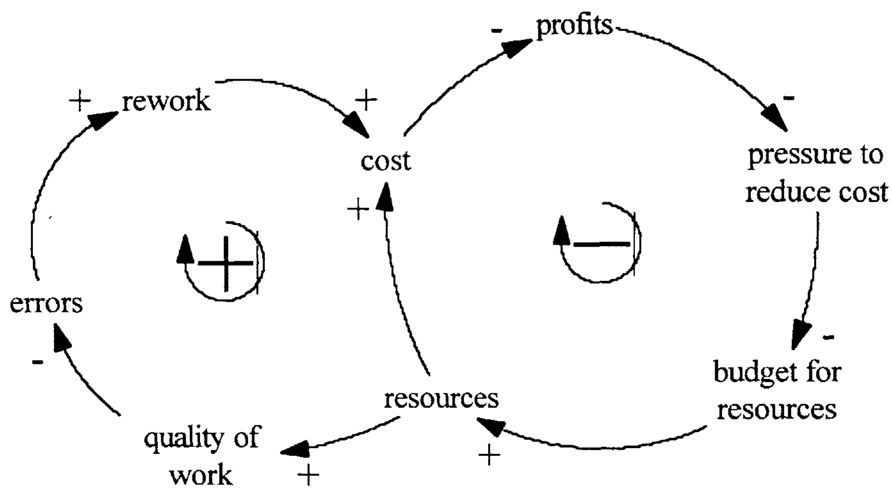


Figure 17: Causal Loop Diagram for "Cost Cutting Through Resource Reduction"

Here, the negative loop will tend to drive cost, and by extension resources, to an equilibrium position that can be sustained by profits. However, if this level is not enough to maintain quality work, cost will be driven up by errors. This erodes profitability and can lead to further need for cost reductions. This could lead to a downward spiral if the company is not careful.

As with hypothesis one, a model was created in Vensim; documentation of the full model and all of its equations can be found in Appendix A. Note that this is a new model, not a continuation of the model built for hypothesis one. The connections and interactions between the two models will be considered later. The first part of the new model, shown in Figure Eighteen, relates to how the pressure to reduce costs can drive the adding and removing of resources.

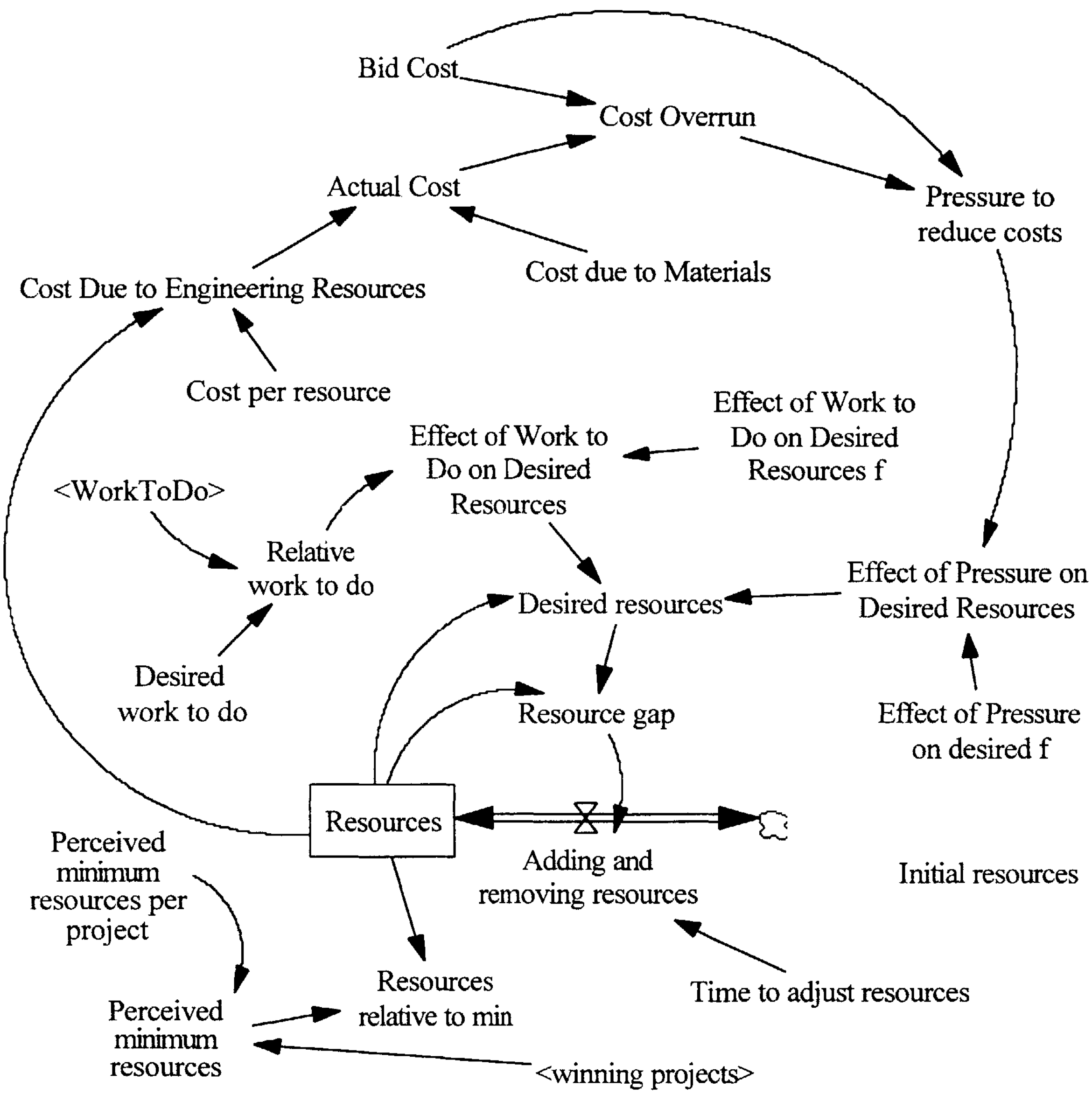


Figure 18: Model for "Cost Cutting Through Resource Reduction", Part One

Following are some of the key equations for this section of the model.

The level of resources is changed by adding or removing resources in response to the gap between the desired and current resource levels:

Resources= INTEG (Adding and removing resources)
Units = people

Adding and removing resources=(Resource gap/Time to adjust resources)
Units = people/Month

Resource gap=Desired resources-Resources
Units = people

The amount of desired resources is calculated using the current resource level with two effects. First is the amount of work to do; this tries to keep the resource level high enough to complete the required tasks. Second is the pressure to reduce costs[5], which pushes the resource level toward a point which is affordable:

Desired resources=Resources*Effect of Pressure on Desired Resources*Effect of Work to Do on Desired Resources
Units = people

The second part of the model shows how work is accomplished. As projects are won, work to do enters the system. These tasks may be performed correctly or incorrectly, which is governed by the quality level. Those that are performed incorrectly are discovered after a period of time, and they re-enter the pile of work to do. The second portion of the model is shown in Figure Nineteen.

[5] The actual cost includes a component for materials cost. This is not used in this model, but is included to allow for further modeling work on the effects of cost reduction pressure on materials cost.

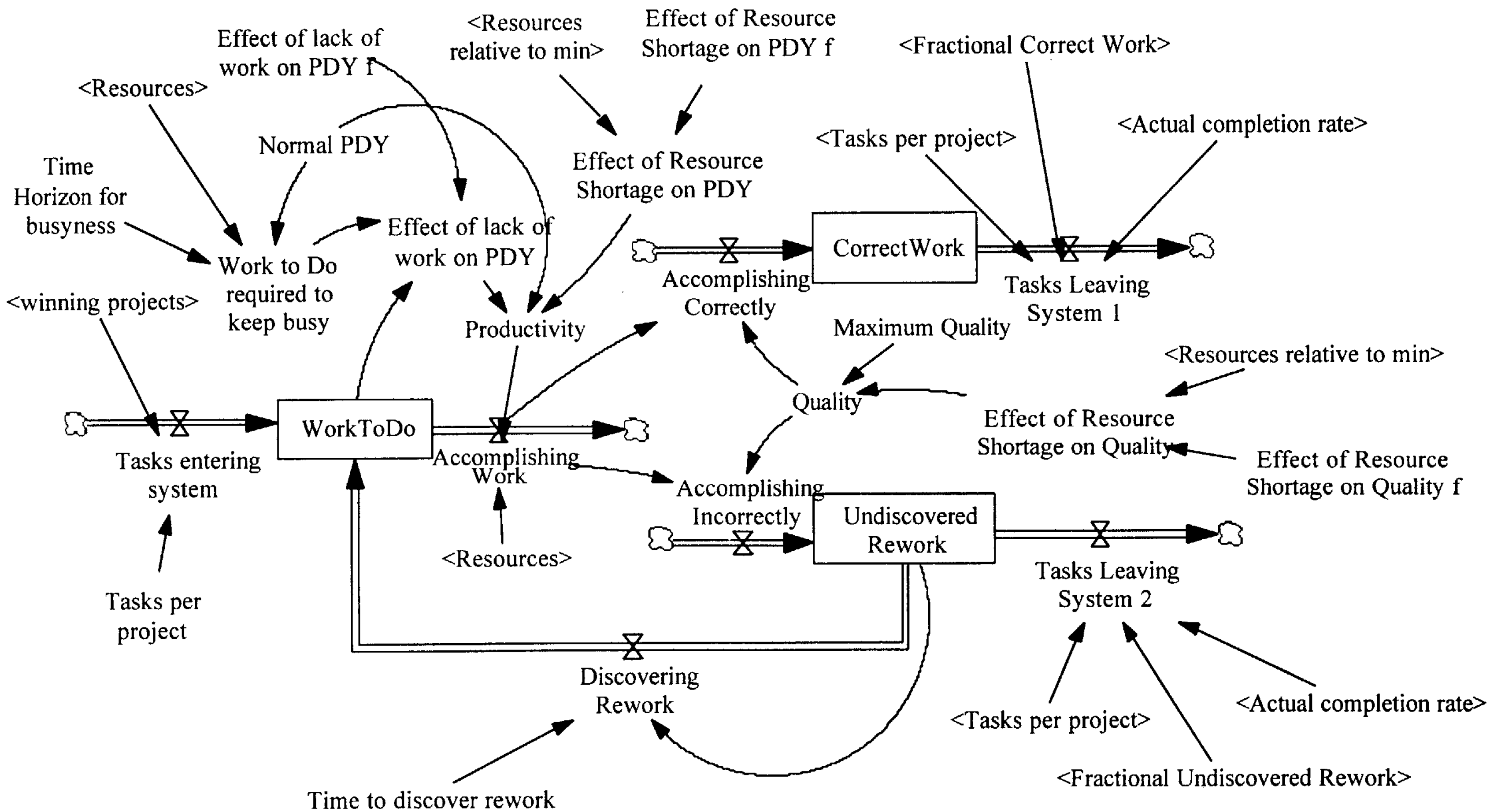


Figure 19: Model for "Cost Cutting Through Resource Reduction", Part Two

The equations in this section revolve around the rate at which work gets done, and how much of it is done correctly[6]. The key equations are given below.

The rate at which work gets done is controlled by the resources that are available and their productivity:

AccomplishingWork= Resources*Productivity
Units = tasks/Month

Productivity can change from its normal level for two reasons. It can increase slightly when there begins to be a resource shortage, since people are forced to take on more work. It can also decrease if there is not enough work available to be done:

Productivity=Normal PDY*Effect of lack of work on PDY*Effect of Resource Shortage on PDY
Units = tasks/(Month*person)

The quality level dictates how much work is done correctly, and how much is done incorrectly:

AccomplishingCorrectly = AccomplishingWork * Quality
AccomplishingIncorrectly = AccomplishingWork * (1 - Quality)
Units = tasks/Month

Quality is influenced by the amount of resources that are available, relative to a minimum level that is needed to work effectively:

Quality=Normal Quality*Effect of Resource Shortage on Quality
Units = fraction

Work leaves the system when a project is complete. Since the participants in the system do not know which work has been completed correctly until rework is "discovered", tasks can leave the system both from Correct Work and Undiscovered Rework. The total rate of projects being completed is

---

[6] This section of the model is based on the "work accomplishment structure", also called the "rework cycle". It is found in many well known project models. This author used as references the molecules in Vensim (a set of commonly found modeling structures) and the teachings of Dr. James Lyneis in the fall 1998 session of the System and Project Management course at MIT.

controlled by the next section of the model. This completion rate is then divided proportionately and converted to units of tasks. For example, the equation for tasks leaving Correct Work, called Tasks Leaving System 1, is:

Tasks Leaving System 1=Actual completion rate*Fractional Correct Work*Tasks per project
Units = tasks/Month

The third portion of the model, which as noted above controls the completion of projects, is shown in Figure Twenty.

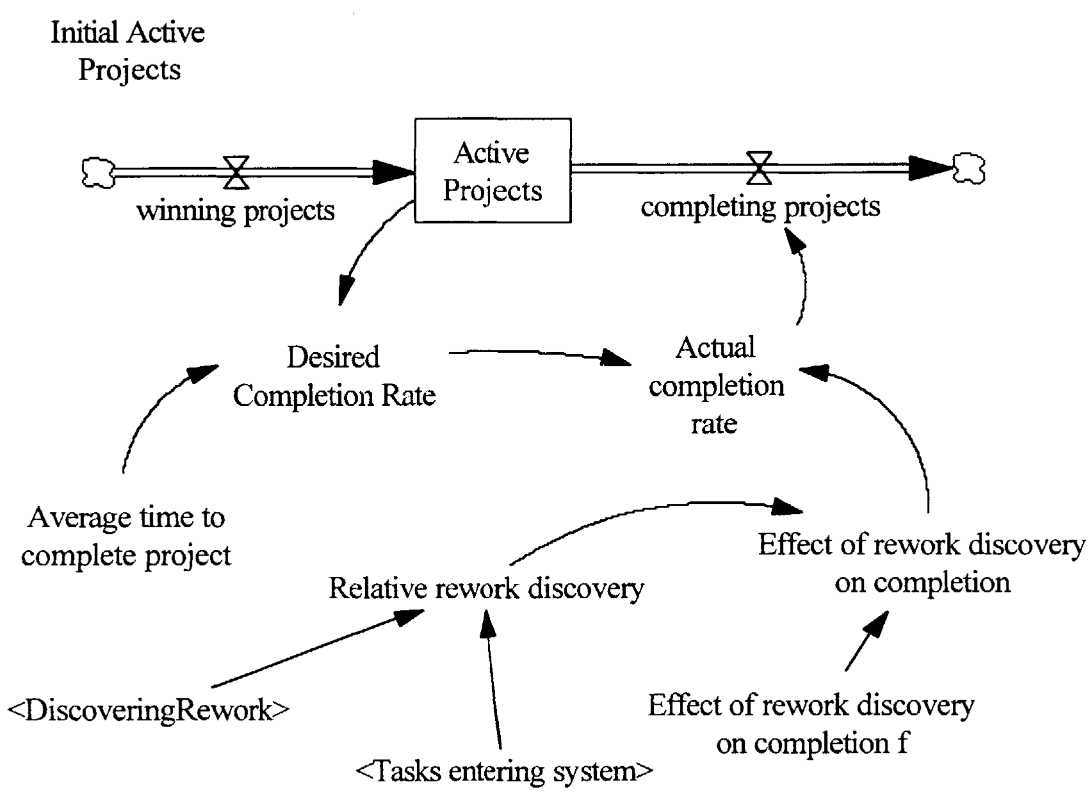


Figure 20: Model for "Cost Cutting Through Resource Reduction", Part Three

The desired rate of completion is dictated by the average time to complete a project:

Desired Completion Rate=Active Projects/Average time to complete project
Units = projects/Month

The actual rate of completion may be reduced if the company feels that the quality of work is slipping. This is judged based on the proportion of work coming in that is rework. If rework being discovered is more than 25% of the total tasks entering Work To Do, ProductCo reduces the rate at which projects are allowed to leave the system.

Actual completion rate=Desired Completion Rate*Effect of rework discovery on completion
Units = projects/Month

Relative rework discovery=DiscoveringRework/(DiscoveringRework+Tasks entering system)
Units = fraction

The behavior of this model is much more difficult to predict than that of hypothesis one. We have two competing pressures - pressure to reduce cost, and pressure to get the work done. It is possible that one of these will be dominant and overcome the other. That is, the pressure to reduce costs could be so powerful that all other considerations are lost. Alternatively, one factor could dominate up to a point, to be superseded by the other when a certain level is reached. This is a strength of system dynamics - to show behavior which is not intuitively obvious due to the complexity of the system.

When initially run, the model quickly reaches equilibrium. Figure Twenty-One shows the behavior of the resource level. It drops to a point which balances the effects of the cost overrun and the need to get work done. The effects are shown in Figure Twenty-Two.

Graph for Resources

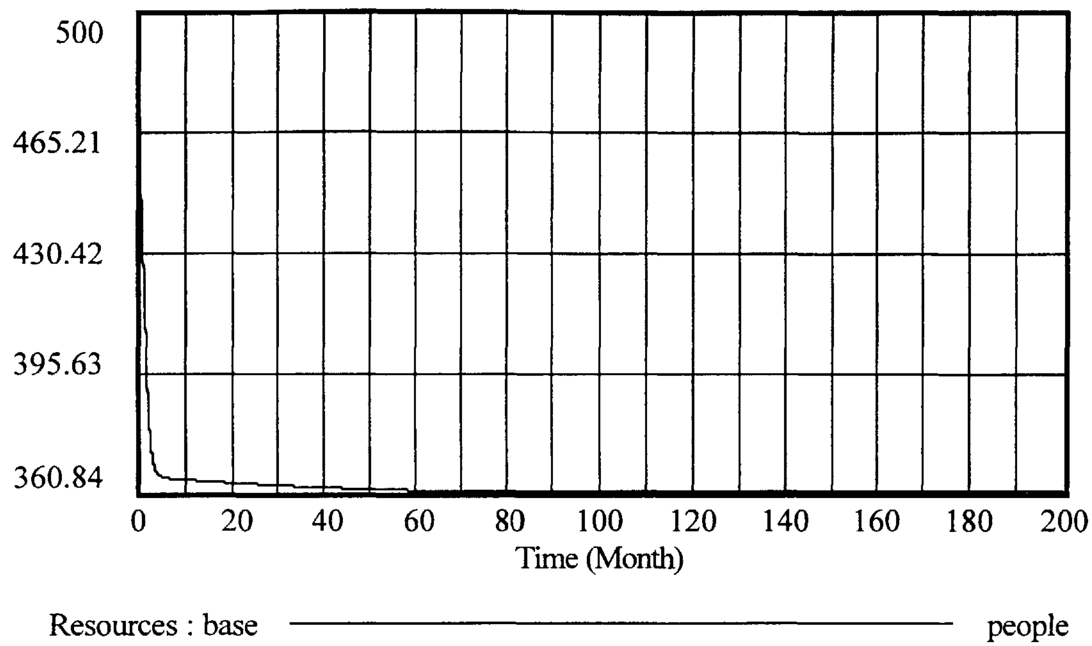


Figure 21

Effect of Price Pressure and Effect of Work To Do

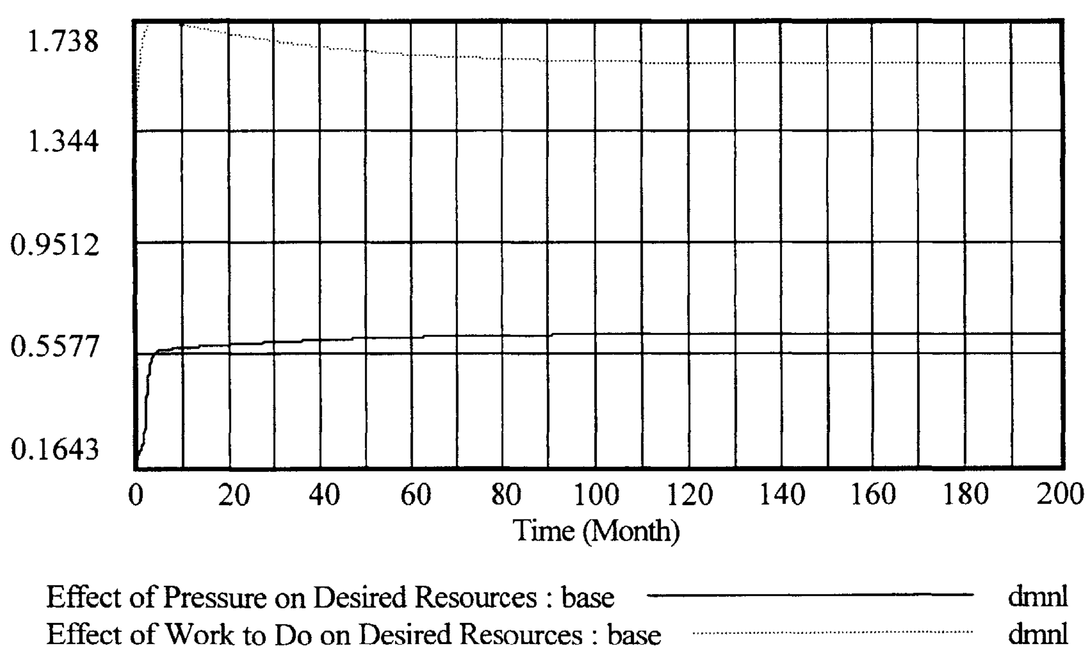


Figure 22

This stabilizes the cost overrun, although it does not eliminate it. The graph for Cost Overrun is given in Figure Twenty-Three. Because, as shown above, there is continuing pressure to keep resources at a level that is adequate to keep Work to Do from piling up, the cost overrun does not reach zero in this scenario.

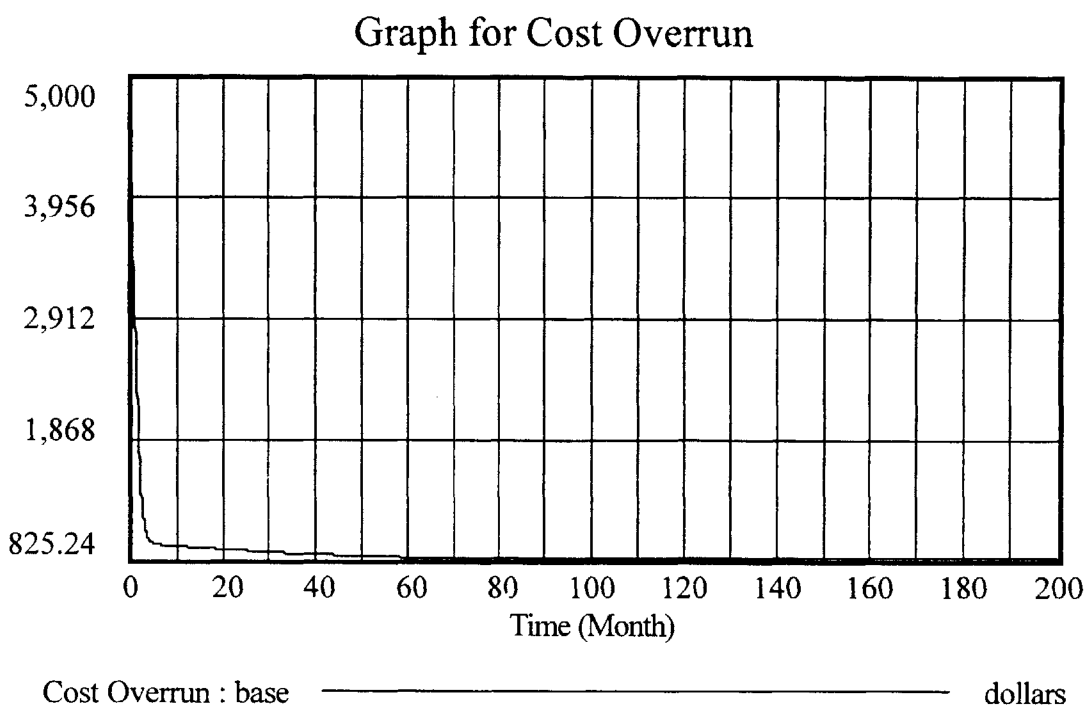


Figure 23

This points out an issue that ProductCo must address. They will need to find ways to reduce costs other than reducing resources. Alternatively, they can try to find ways to get more out of the resources that they have. This will be discussed in more detail later in this section.

As pointed out, keeping resources at a certain level allows work to move smoothly through the system. Work to Do is shown in Figure Twenty-Four. It initially grows as resources are cut, then levels off as resources stabilize.

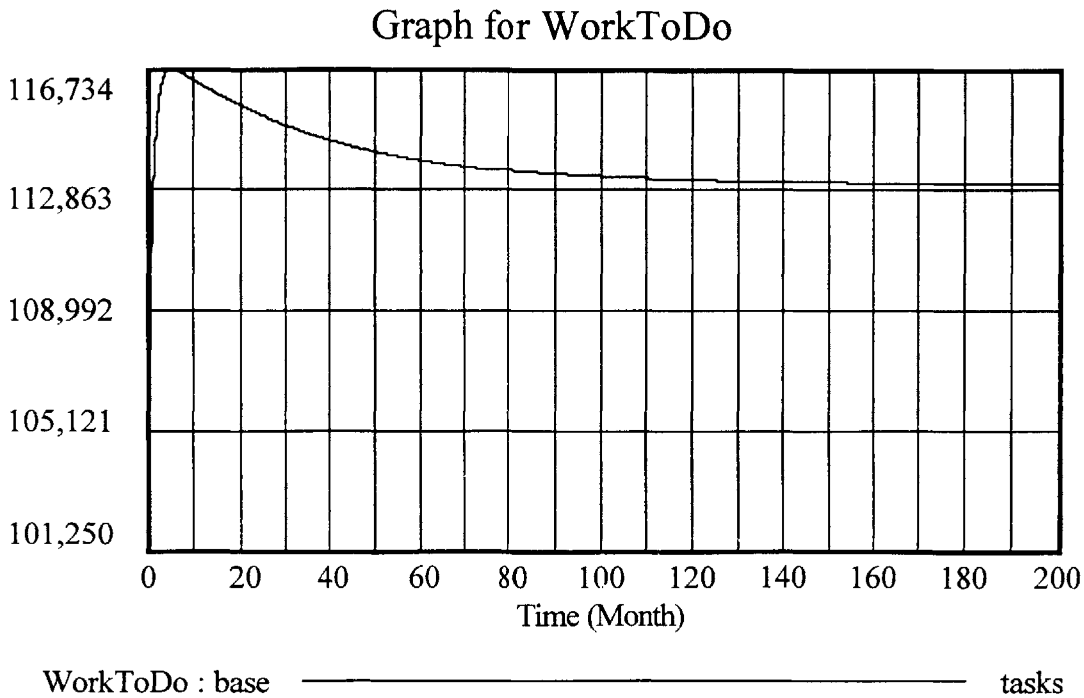


Figure 24

One surprising aspect of the Work to Do graph is that it decreases after it reaches its peak. This occurs because the rate of accomplishing work remains above the total inflow (Tasks Entering System plus Discovering Rework) until the equilibrium resource level is reached. This tells us that in the real system, if the backlog of Work to Do is to be reduced, then the rate of work getting done must be greater than the work coming in. This is fairly obvious; however, what might not be obvious is that we have to consider the flow of rework being discovered as well as the new work entering the system.

Further analysis was performed on the model by placing it in equilibrium, then exciting it by adding an input that pushes it out of its equilibrium state. The response to this change can yield some insights into the behavior of the real world system. In this case, the rate of projects was stepped up by one per month. The added model structure is shown in Figure Twenty-Five. When the switch is turned "on" by setting its value to one (normal value is zero), it activates a step function which increments the rate of winning projects by one, starting at time 500.

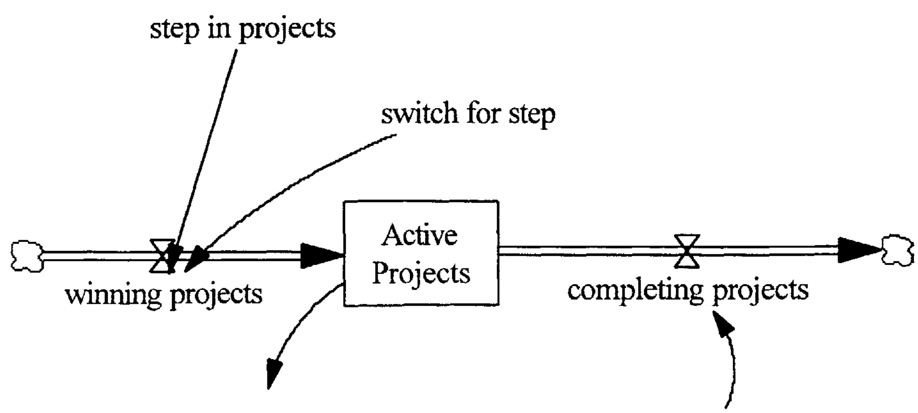


Figure 25

This change creates an increase in the work to be done. This, in turn, increases the required resources. The behavior of the stock of Resources is shown in Figure Twenty-Six. It rises to a new, higher equilibrium. The interesting aspect is that it does not rise at a steady rate. Note that the Resources curve is exactly the same shape as the Cost Overrun curve, since the cost in this model is controlled only by resources.

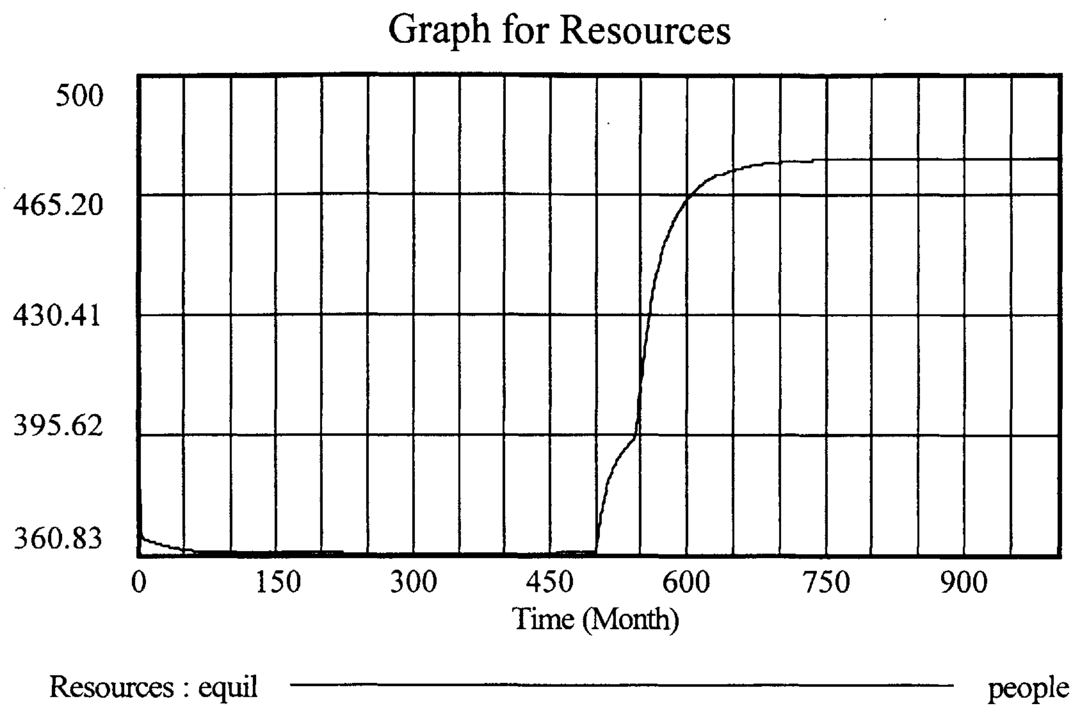


Figure 26

This change in slope is a result of the effects that are impacting the Desired Resources. One possibility is that one effect is dominating, but is then overcome by the other. However, this is not the case; if it were, a change in the direction of the resource curve would be likely since the effects are pulling in opposite directions. Figure Twenty-Seven shows the resource curve overlaid with the effects. Another possibility is that the changing slopes of one of the effects is causing the discontinuity in the Resources curve.

Resource Level vs. Effect of Price Pressure and Effect of Work To Do

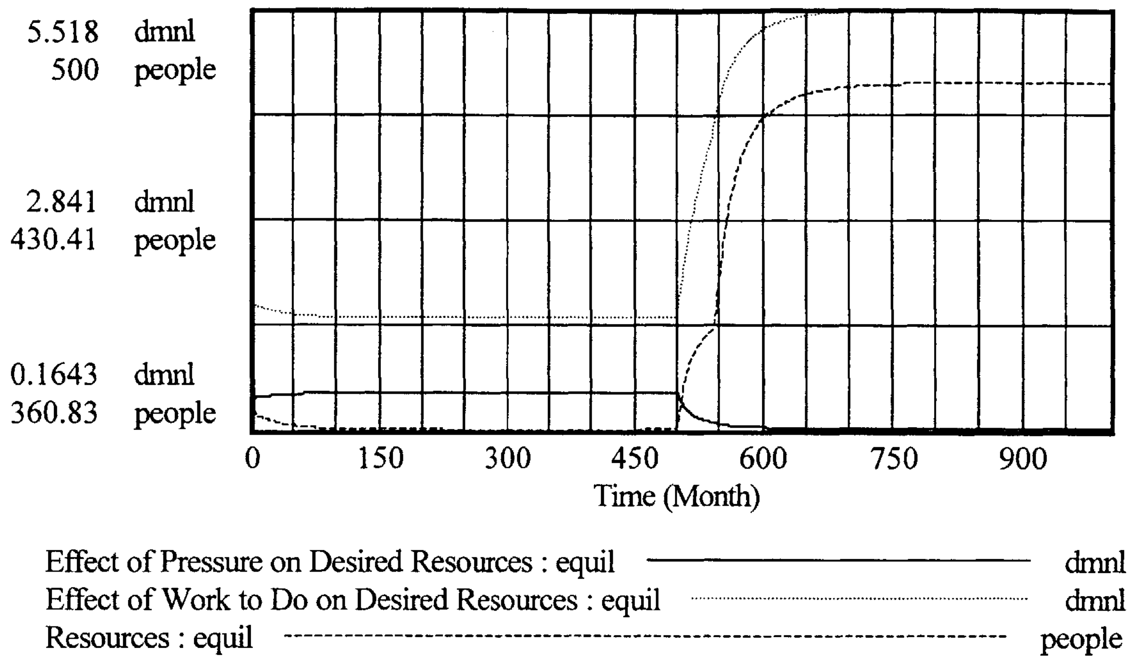


Figure 27

Both of the effects are calculated using table functions. These are shown in Figure Twenty-Eight, with the table function for Effect of Work to Do on Desired Resources on the left, and the table function for Effect of Pressure on Desired Resources on the right.

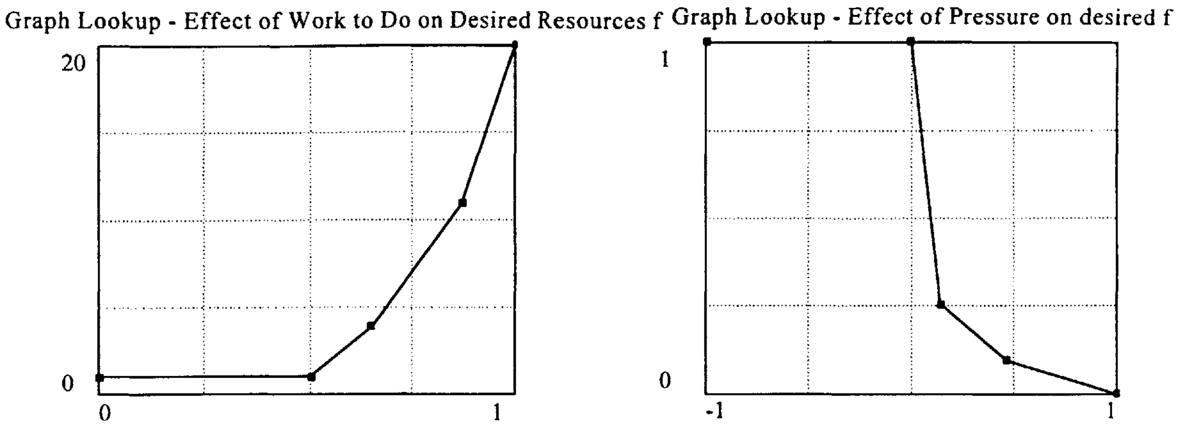


Figure 28: Table Functions

In order to determine whether the changing slope of one of these functions was causing the shape of the Resources curve, simulations were run which removed points from each table function, rendering them more linear. Points were removed between x = 0.5 and 1.0 in the case of the effect of work to do. The results of these simulations are shown in Figure Twenty-Nine. The table for Effect of Pressure on Desired Resources was reset to its original shape, and points were then removed between x = 0 and 1.0 in the table function for the effect of pressure. The results of these simulations are shown in Figure Thirty.

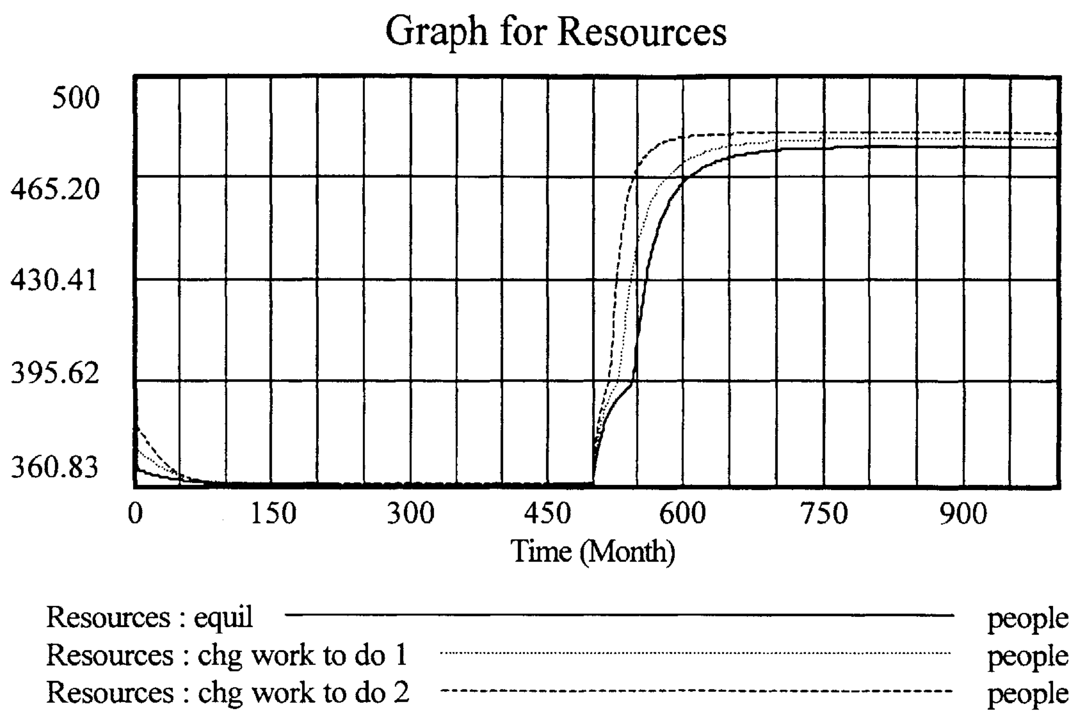


Figure 29

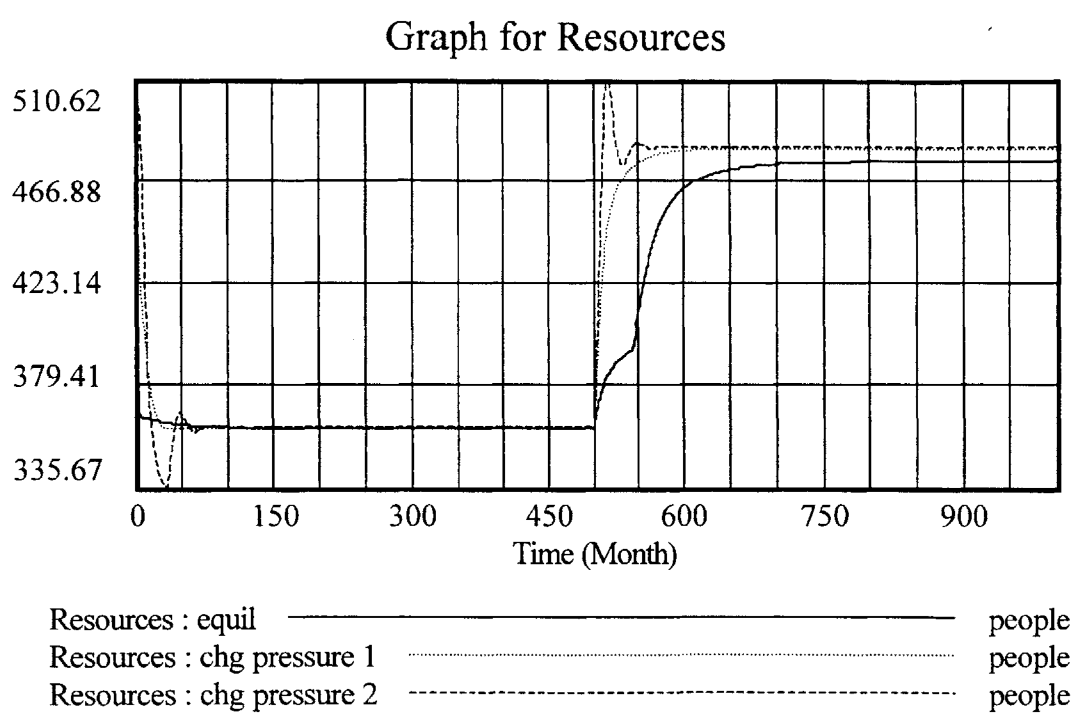


Figure 30

Changing the shape of the table function for the Effect of Work to Do on Desired Resources does not change the shape of the Resources curve. It only shifts it slightly, reaching equilibrium sooner. This is because as the table function is made more linear, it causes the system to react faster to the buildup of work to do. It pulls the resources up more quickly in response, and it reaches a higher equilibrium level of resources since the counteracting effect due to price pressure has not changed.

On the other hand, changing the shape of the table function for the Effect of Pressure on Desired Resources has a more dramatic effect on the shape of the Resources curve. It first removes the step in the curve, and then causes an overshoot and recovery pattern. When the table function is completely linear between x = 0 and 1.0, the resource level overshoots and then oscillates briefly before reaching its equilibrium level, which is higher than the original equilibrium level. This occurs because removing the points from the table function causes the system to react more slowly to the rising cost overrun. It allows resources to build in response to the work to do, and then works to bring them down when the cost overrun gets too high.

This analysis points out that if the company does not react strongly enough to cost overruns, the eventual problem will be worse. In the original setup, the Effect of Pressure on Desired Resources, which is a multiplier on Desired Resources, dropped off quickly as the Actual Cost moved above the Bid Cost. The fact that the exact points used in the table function caused a step in the curve for Resources is less important; a smoother table function could eliminate this issue. The biggest insight for ProductCo is that they must respond quickly to cost overruns if they want to keep the equilibrium level of cost to a minimum.

There is another issue to be considered, however. If ProductCo reacts too harshly to cost overruns, they may damage their quality. The table function can be changed by shifting points, rather than by eliminating them, to show a quicker response and a slower response to cost issues. The resulting Resources curve is shown in Figure Thirty-One.

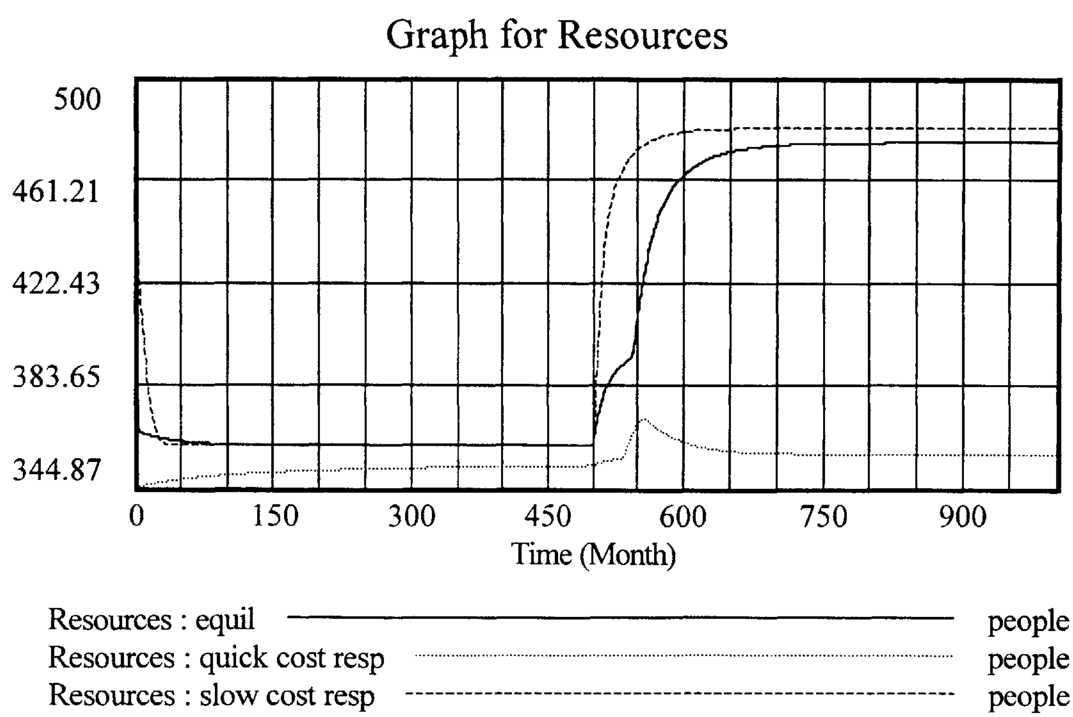


Figure 31

Responding quickly to cost keeps the cost overrun lower, while responding more slowly allows it to become higher. It would seem that ProductCo should follow the policy which produces the lowest cost overrun. But this creates other problems; the graph for quality is shown in Figure Thirty-Two. Since there is an effect on quality which is strongly correlated to the resource level, the quality is also at its lowest point when the company responds harshly to cost overruns.

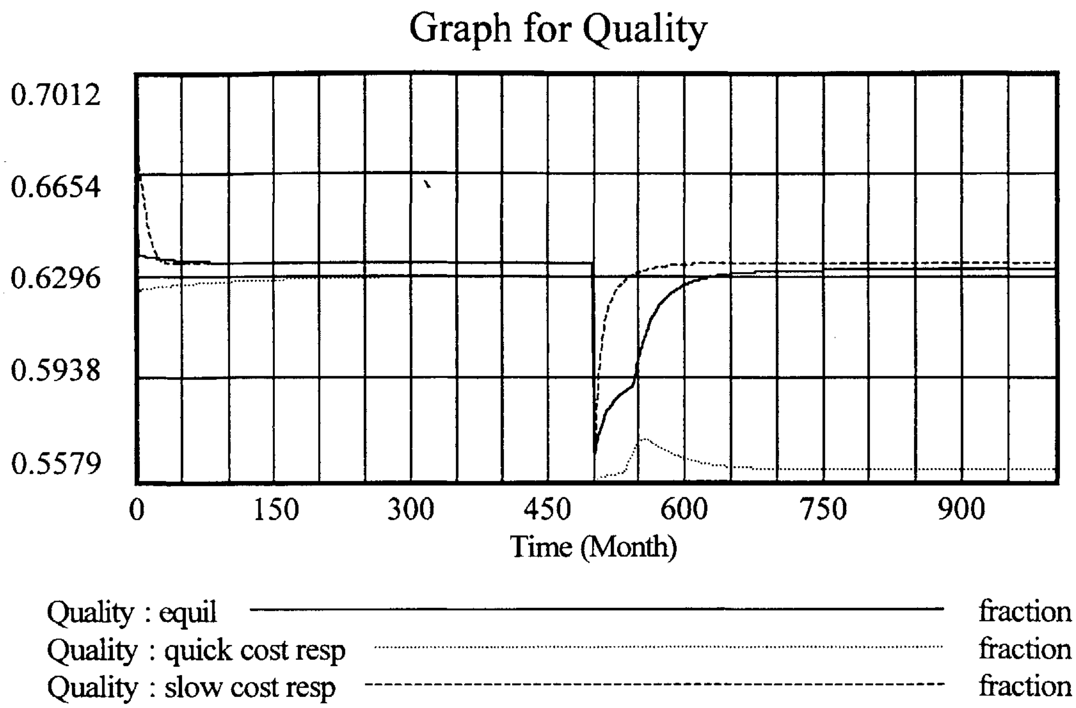


Figure 32

Note that quality takes an initial hit at time 500 because although work has been added, resources cannot be added instantaneously. However, in the original formulation or with a slow cost response, it recovers to approximately its previous level. With a quick cost response, resources remain short and quality does not recover.

One could predict that this would lead to excessive work to do, since the rework would be increased if the quality is lower. The work to do pile does reach a higher level when the cost response is faster, but it is not due to rework. Figure Thirty-Three shows the graph for Work to Do, while Figure Thirty-Four shows the rate of Discovering Rework.

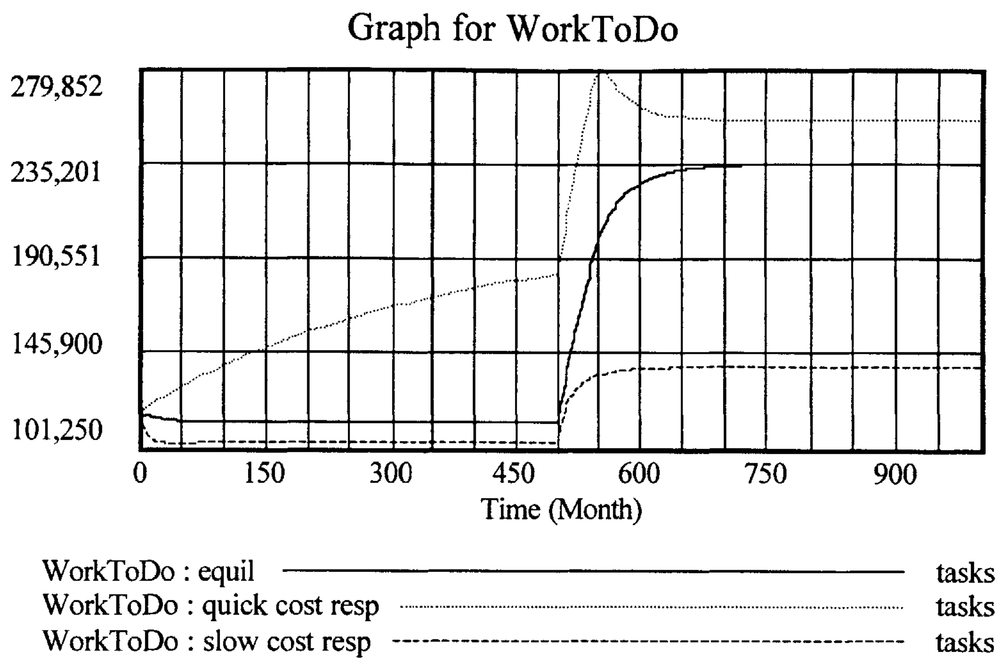


Figure 33

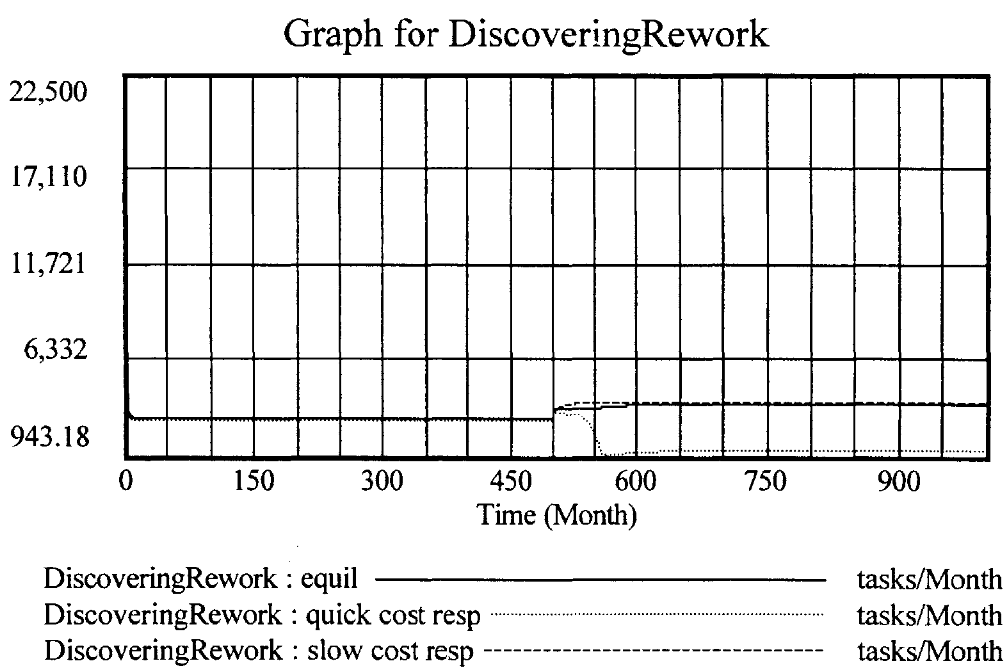


Figure 34

This shows that with the slower cost response, the rate of rework discovery does not change significantly, and it actually drops slightly with the quicker response. How is this possible? As it turns out, with the quicker cost response, the work is flowing more quickly into Undiscovered Rework, but it is being shipped out the door before it is discovered.

Comparing the quick cost response case to the original equilibrium case, we can see why this occurs. If the quality does not recover fairly quickly, the proportion of work that is flowing into rework remains high. This raises Fractional Undiscovered Rework, which compares the level of undiscovered rework to the total work perceived to be complete (Correct Work plus Undiscovered Rework). The graph for Fractional Undiscovered Rework is shown in Figure Thirty-Five.

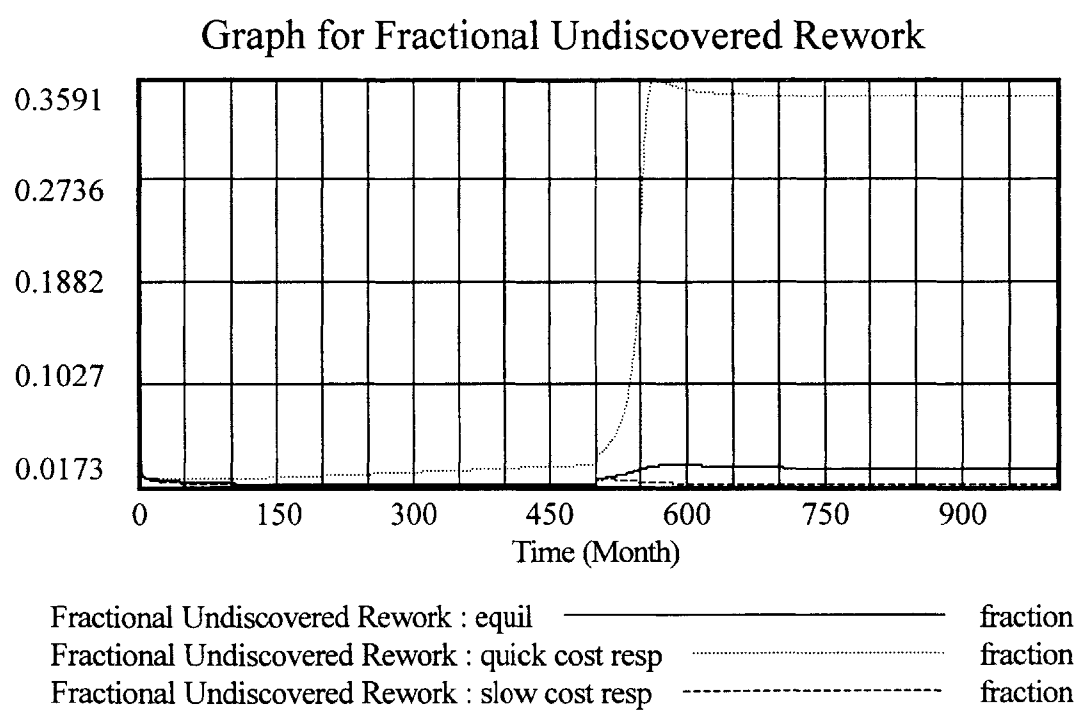


Figure 35

The large jump in Fractional Undiscovered Rework occurs at the same time that the overall completion rate increases. When the incremental project was added, the desired completion rate for projects increased. Projects are expected to be completed at the same rate despite the fact that resources cannot be added instantaneously. These two factors cause an increase in the rate of tasks flowing out of Undiscovered Rework, called Tasks Leaving System 2. This is shown in Figure Thirty-Six.

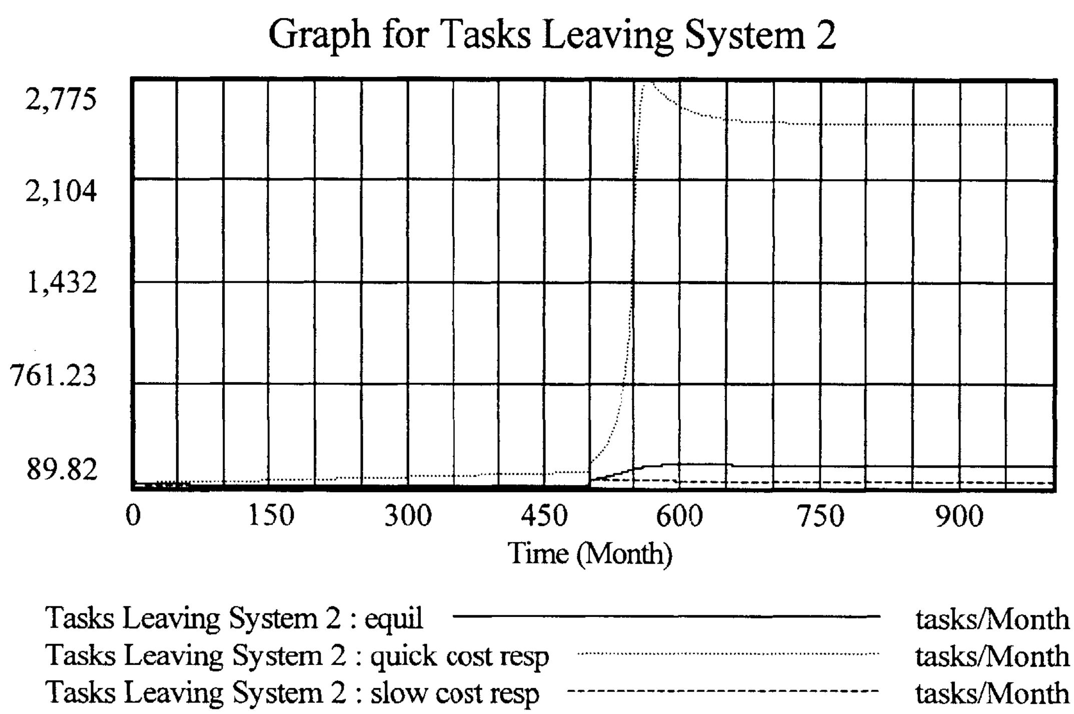


Figure 36

Since Undiscovered Rework begins leaving the system faster at about the same time that it starts being created faster, the net impact on the level of Undiscovered Rework is a minimal change. The impact on ProductCo could be much greater. With the quick cost response, they are shipping projects with many more errors. This low quality level will not be well received by their customers. This would likely result in a lower rate of winning projects over time. This can be addressed when hypotheses one and two are combined.

If it is not due to rework, the buildup in Work to Do must be attributable to some other factor, namely the rate of accomplishing work. Because the quicker cost response limits the amount of rework that can be added, it also limits the amount of work that can be completed in a given time period. Productivity actually increases slightly with the quicker cost response. People can increase their productivity in response to a resource shortage, but only to a certain point. And, as noted previously, quality of that work will suffer. The graph for Productivity is shown in Figure Thirty-Seven.

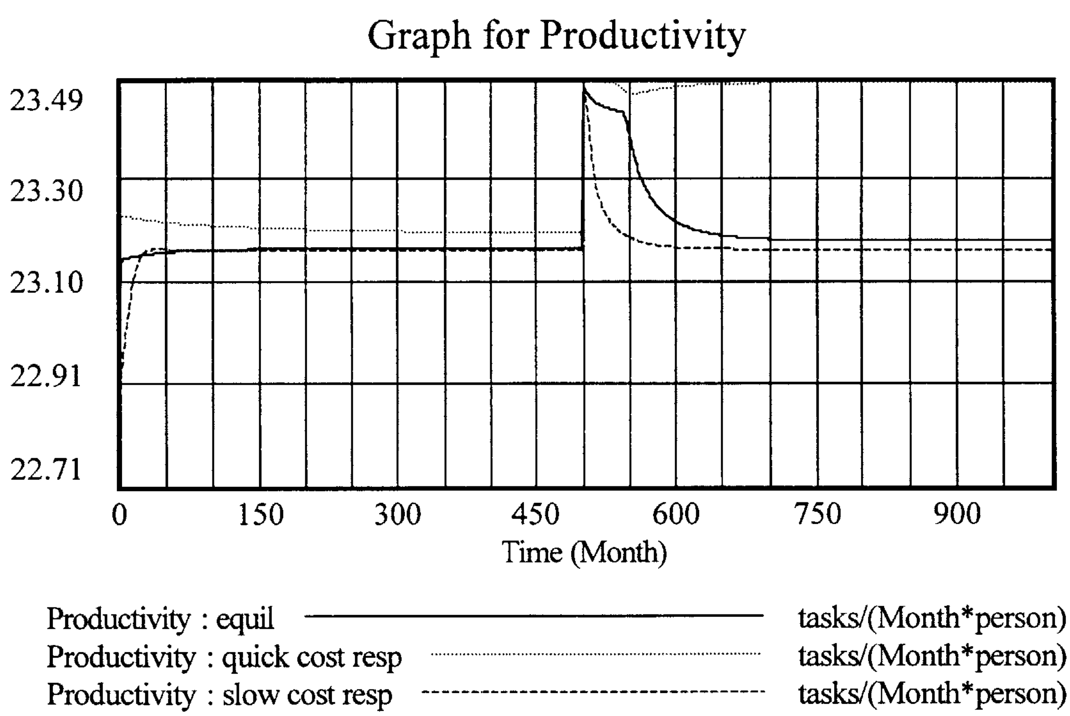


Figure 37

However, this cannot make up for the lack of resources. The graph for Accomplishing Work is given in Figure Thirty-Eight. Since work cannot flow out of Work to Do as quickly, the backlog builds up to a higher level, as was seen in Figure Thirty-Three.

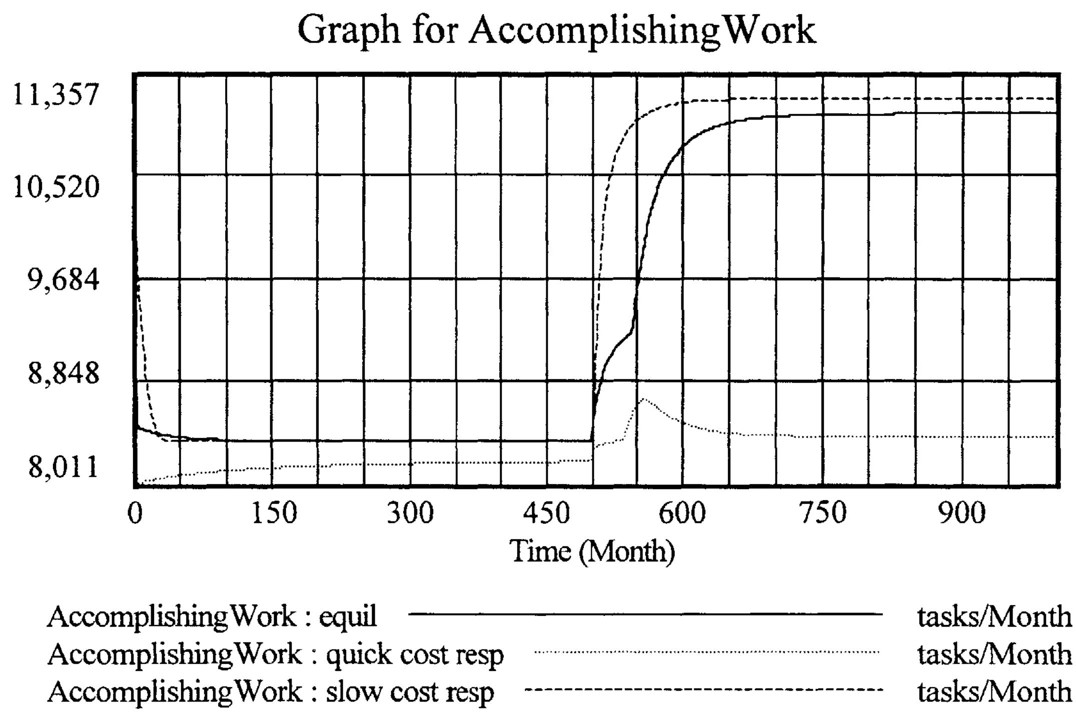


Figure 38

This buildup of work to do does not affect the rate at which projects are being shipped out the door in this model, although with the quick cost response they are being shipped with many more errors. However, it may make it more difficult to get a high-priority project pushed through the system quickly, since work is accomplished fairly slowly and there are many tasks waiting to be addressed.

The equilibrium analysis of this model yielded several insights for ProductCo. The response of the company to a cost overrun is vital. A strong response can ensure that the cost overrun is limited.

However, this has serious repercussions on quality. The analysis also shows that this quality problem may be hidden by the fact that the errors are being shipped out before they are discovered. If ProductCo uses relative rework discovery as an indicator of when it is okay to ship, this problem could begin to effect customer perceptions before ProductCo is even aware of it. Finally, with limited resources, work cannot move through the system quickly, which could cause issues if a high-priority project must be expedited.

Using the model, I also looked at which factors have the biggest impact on the dynamics of the system when they are changed. The model is very sensitive to the values used for Cost per Resource and for Normal Productivity. This makes sense, as both of these parameters directly affect how much work we can get for every dollar we spend on resources.

By varying the Cost per Resource from its base value of 30 to a lower level of 25 and a higher level of 35, we can see how this affects the dynamics of the model. The graph for Resources is shown in Figure Thirty-Nine. As expected, a higher cost leads to a lower resource level, but the shapes of the curves are not the same.

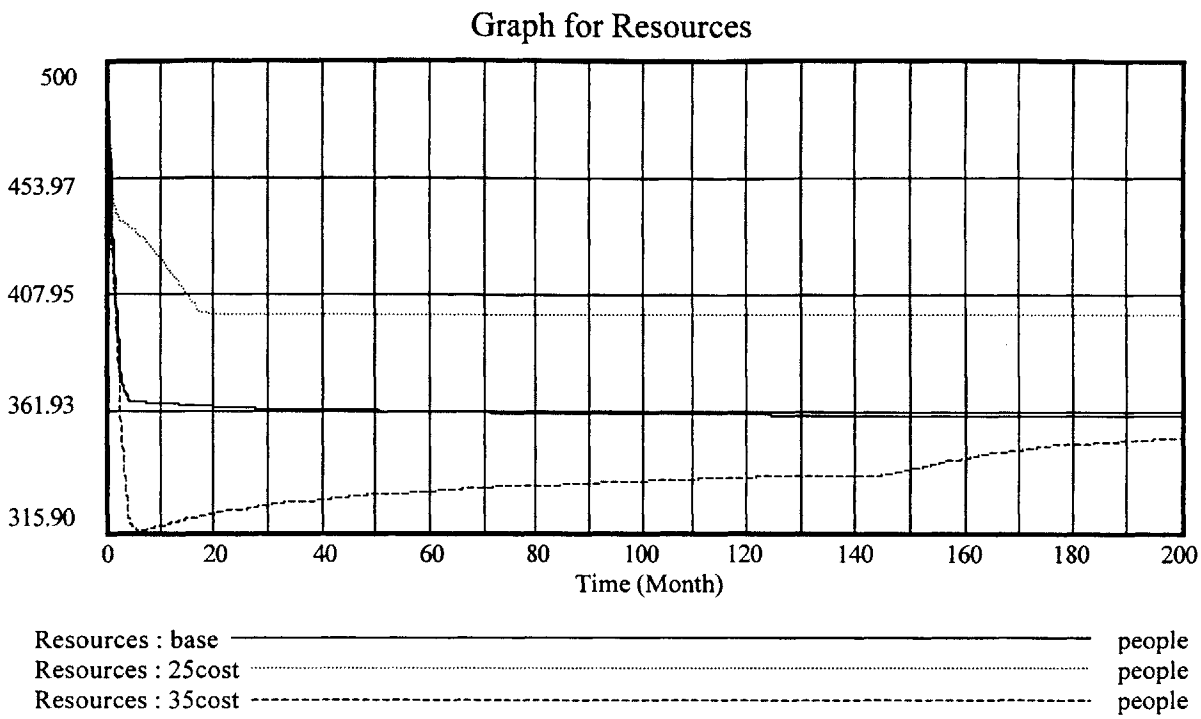


Figure 39

The lower cost curve starts to follow the base curve's track, but then begins to drop more slowly to its equilibrium level. In this case, Pressure to Reduce Costs (based on the cost overrun) quickly drops below 0.1, where its effect on Desired Resources begins to lessen. This is a consequence of the shape of the table function for Effect of Pressure on Desired Resources.

The higher cost curve drops off quickly in response to price pressure, but then rises to approach the base level of resources. This is in response to the buildup of Work to Do as a result of the resource shortage. The graph of Work to Do is given in Figure Forty.

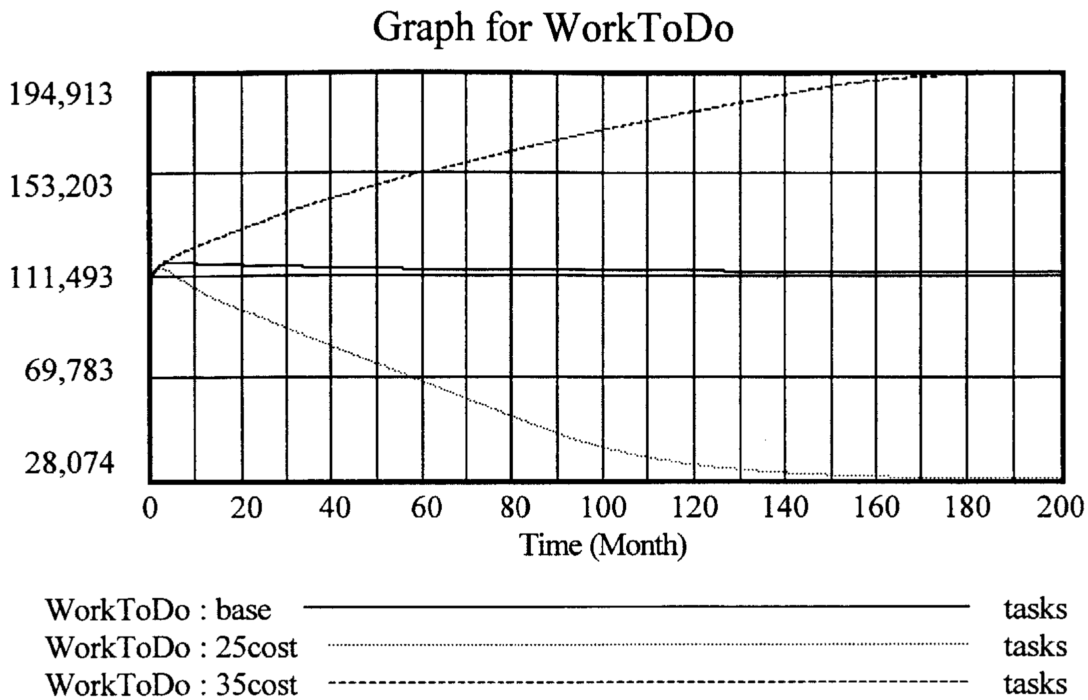


Figure 40

In the higher cost case, we can see here that Work to Do builds until the point that the Effect of Work to Do on Desired Resources becomes stronger. Again, this is attributable to the shape of the table function. At this point, resources are added and Work to Do levels off.

It is important to note that Cost Overrun no longer follows the shape of the Resource curve, since we are changing the relationship of resource level to cost in this set of simulations. The graph for Cost Overrun is shown in Figure Forty-One. Since the higher cost per resource scenario reaches a similar resource level to the base run, the cost overrun is much higher.

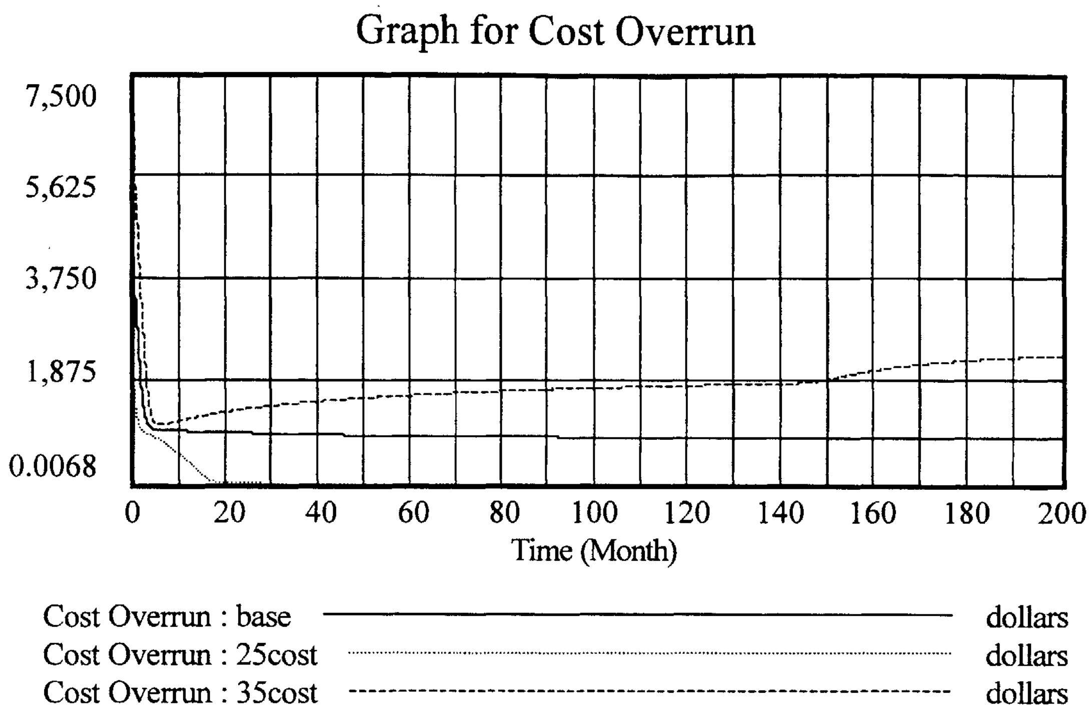


Figure 41

Cost per resource naturally rises over time with inflation. This can be exacerbated by heavy competition for talent in an industry, which may drive wages up more quickly. As this analysis shows, ProductCo must be alert to these changes and must price for them in order to avoid higher cost overruns, and resource reductions which cause work to slow down. They could also look for ways to reduce their cost per resource. This would be most effective if they can do this without decreasing wages, such as negotiating better rates for benefit plans or decreasing overhead.

On the flip side, the company can also benefit from increasing the amount of work that each resource can accomplish. This is measured by productivity. Similar to the analysis performed on Cost per

Resource, we varied Normal Productivity from its base value of 20 to a high of 25 and a low of 15. The resulting Resource curves are in Figure Forty-Two[7].

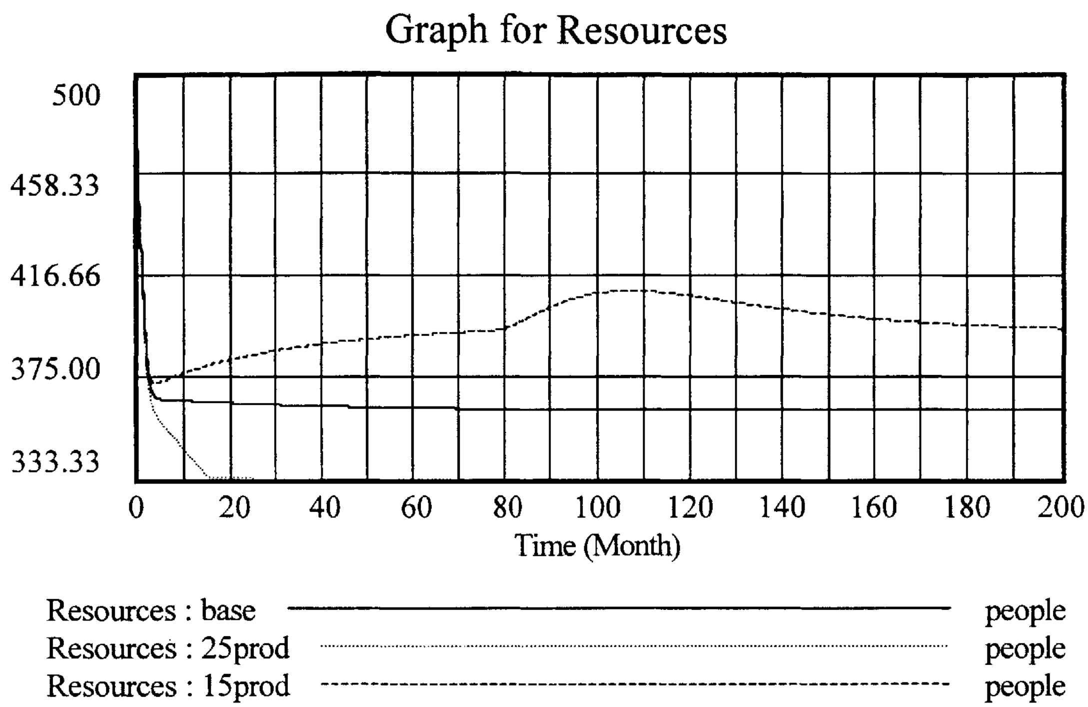


Figure 42

The high productivity run behaves much the same as the low cost scenario previously discussed. Resources drop off in response to price pressure; the rate slows down as the cost overrun approaches a low level. However, the low productivity model produces the surprising result of peaking and then decreasing resources.

This occurs as a result of the effects on Desired Resources. The Effect of Work to Do on Desired Resources reaches a point where it becomes stronger just as the Effect of Pressure on Desired Resources begins to slow down. This allows the resource level to increase quickly, which overshoots

[7] In this case, Cost Overrun mirrors Resources since the relationship between the two is held constant across all runs.

its goal. There are then more resources than are required to do the work; as the pile of Work to Do is depleted, the resource level is pulled back down to an equilibrium level. The effects overlaid with the resource level for the low productivity scenario are shown in Figure Forty-Three.

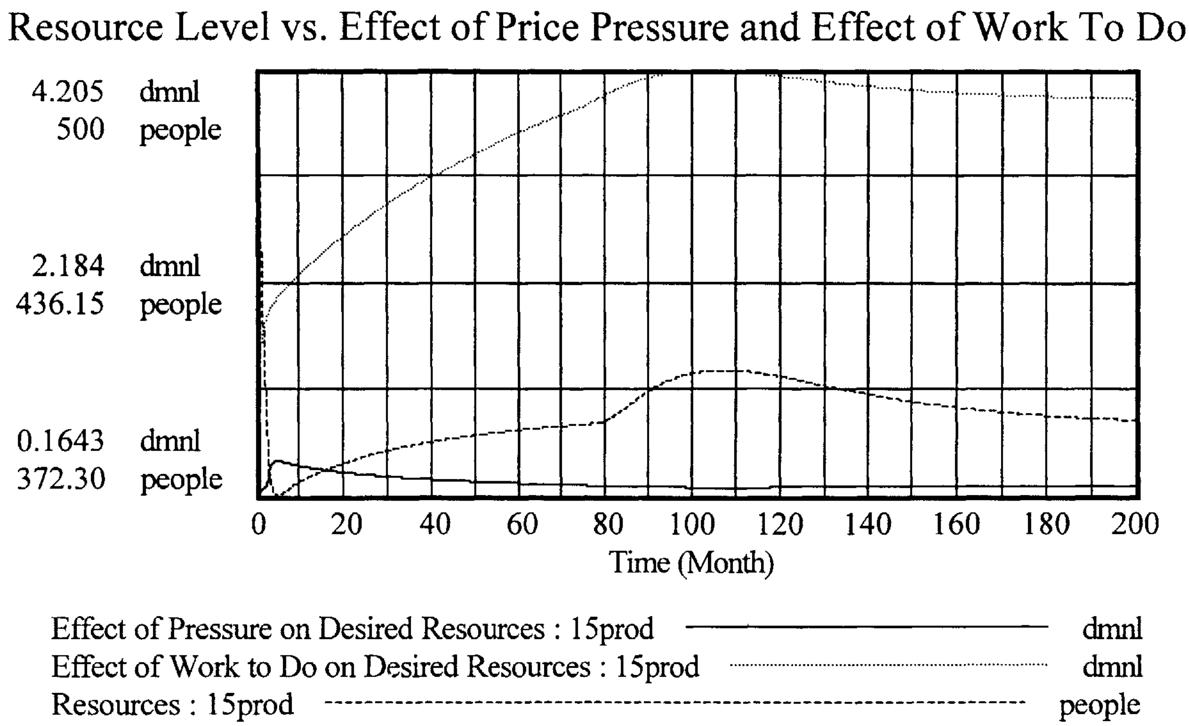


Figure 43

Lower productivity also has the effect of causing more mistakes to be shipped out in completed projects, similar to the effect of tight cost control. The low level of productivity prevents work from moving quickly through the system. At the same time, projects are expected to ship on schedule. This depletes the stock of Correct Work, raising the proportion of Fractional Undiscovered Rework. More tasks flow out of Undiscovered Rework (Tasks Leaving System 2) with low productivity than in the base case, as can be seen in Figure Forty-Four.

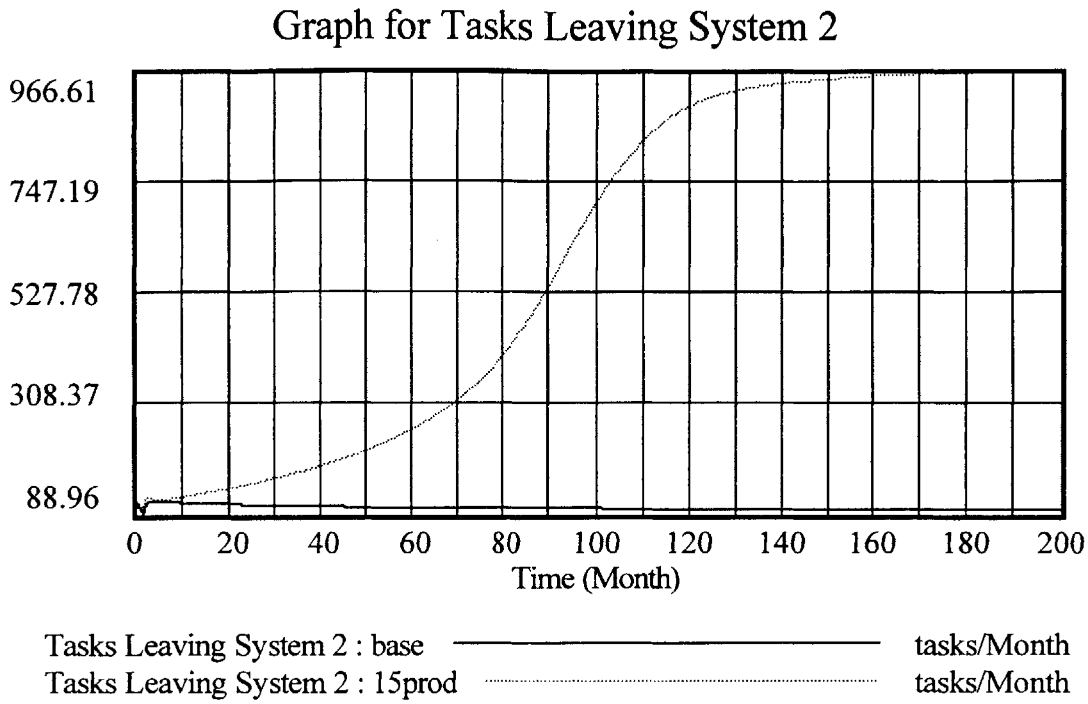


Figure 44

This shows that if productivity slips at ProductCo, they can expect the need for higher resources, resulting in higher cost overruns. They can also expect that defects that reach the customer will rise, which will not be good for the company's reputation. This is a strong incentive to pay attention to factors such as morale which might impact the productivity of ProductCo's resources.

They can also see that higher productivity can enhance their ability to complete their projects within cost. This shows that investments in methods for increasing productivity would be worthwhile. Working smarter, not harder, would be a big benefit to the bottom line. This is easier said than done in most cases, but this analysis shows why it is an important thing to consider.

Many insights were gained by analyzing the models for both hypotheses. The next step is to put the two together and look at the interactions between them. This can provide further understanding of ProductCo's system, and may help point the way to policies that will improve it.

Putting It Together

The causal loops for the combined hypotheses are given in Figure Forty-Five. This shows our overall theory of what is driving profitability at ProductCo. The key connections between the two hypotheses are through cost and price. Hypothesis One drives the price that is asked for a project by the desired margin and the competitive price. Hypothesis Two drives the cost through the resources that are required to complete the projects.

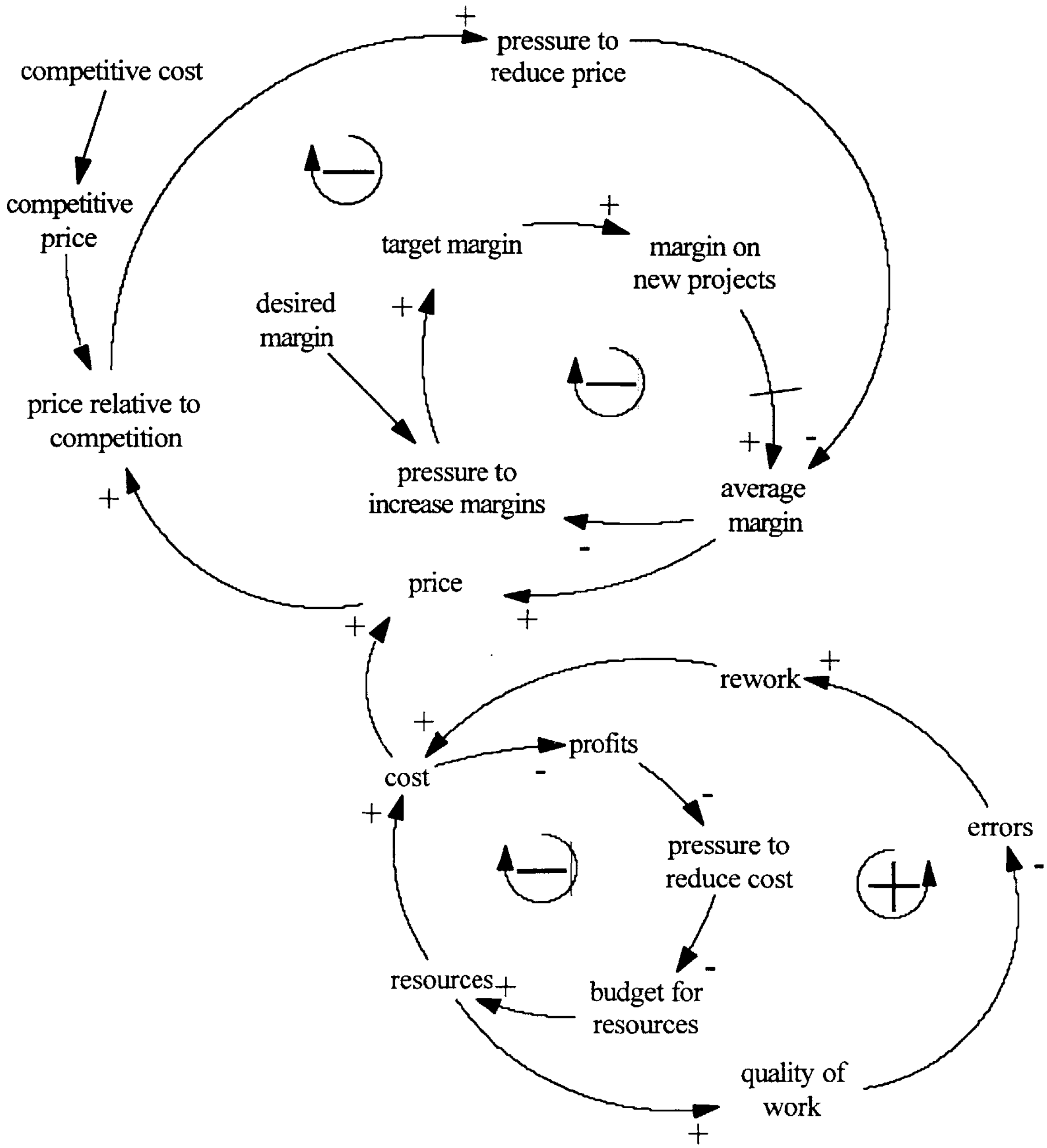


Figure 45: Combined Causal Loops

As mentioned previously, there is also a link through the quality of the projects that are being shipped to the customers. This will impact the win percentage for new projects. This link will provide the ability for the model to "go out of business". That is, a downward spiral could occur with too few resources causing low quality, which decreases the amount of business won. This reduced income could cause further resource reductions, and so on. This will be an important avenue to pursue.

The combination of the two models is beyond the scope of this thesis. It is suggested for future work by ProductCo in order to continue the study of the causes of their profitability issues.

Exploring Potential Policies

What can be learned from this modeling and analysis of ProductCo's system? The goal of the project is to help the company both understand what's going on, and improve its profitability.

In the analysis of Hypothesis One, we saw how important competition is in determining how much can be charged for a given project. It is key for ProductCo to understand what the market price is for its products. It is also vital that they understand differences in items such as overhead or labor rate that might drive differences in their costs compared to the competition's.

Hypothesis One also showed that there is internal tension between management goals and realistic prices. If this cannot be relieved by increasing prices, it must be recognized and handled through other means if ProductCo is to reach its objectives.

Hypothesis Two gave some insights into the pros and cons of dealing with this pressure by cutting costs. Specifically, it focused on resources. In this model, a cost overrun was incurred despite the pressure to cut costs. This was due to the competing pressure to keep enough resources to get the job done. ProductCo must balance these two needs in developing policies which will help it reach its profitability goals.

Further analysis showed that if ProductCo responds too strongly to the pressure to cut resources in order to reduce costs, it could end up with serious productivity and quality problems. Not only will there be a high backlog of work to be done, but more errors will reach the customer. This is certainly not a desirable situation.

ProductCo might also want to re-examine the use of the rate of rework discovery as a means of controlling the shipping rate. It is clear that this can be deceiving. However, it is difficult for them to know what the actual quality level is, since rework is not known until it is "discovered".

Finally, productivity and cost per resource were determined to be key factors in the smooth operation of the product development process. These are key leverage points for ProductCo to focus on, both in trying to improve the system and in trying to prevent undesirable scenarios.

The model and its analysis are not complete enough to recommend full-blown policies at this point. The insights to date can give some guidance, and have helped ProductCo to better understand the system. This can help them in making some policy decisions, even at this early stage.

Conclusions

The objective of this project was to determine whether system dynamics could help ProductCo to improve its product development process. After zeroing in on profitability as the key issue, we attempted to provide a better understanding of the system through causal loop diagrams and modeling, and to use these insights to develop new policies.

I think we have succeeded in gaining insight on the current system and thinking about ways to improve it. We have fallen short of actually developing and testing new policies to improve the system. More work remains to refine the model before this step can be taken.

Some issues were encountered over the course of this project that prevented it from progressing more quickly. These may be common to other system dynamics projects and thus warrant discussion here.

Just after this project was started, ProductCo decided to reorganize its product development process. They added a new position called the Program Manager, who is responsible for the delivery of a given project to the customer. This person is charged with managing the engineering team, financial and business planning components, and overall quality and customer service. This is intended to provide one point of contact for the customer. It is also intended to help keep costs in line with those used in developing the bid, since the Program Manager will be responsible for meeting the financial targets of the project.

It is as yet too early to tell how this will affect the current system. It also took a lot of the time and attention of the people that were needed to work on this thesis project. Since those most knowledgeable about the current product development process were those who were put to work in developing the new organization, it slowed down the progress of this project and may have made the ultimate model less accurate due to more assumptions made for lack of input.

This can happen in any improvement effort; the person driving the project may be reassigned to another part of the company, or a total reorganization can shift the focus. In this case, it has slowed down the project, but hopefully elements of the new organization can be incorporated into the model and it can be used to further improve the new system.

Furthermore, it was found that a strong advocate of the system dynamics method would have helped to move the project along. ProductCo has implemented some employee training using the learning organization concepts as presented in the Fifth Discipline[8]. Some in the company have embraced the idea of systems thinking and are very interested in taking it further to formal modeling. Others, however, have seen little follow up after the training and have dismissed it as just another management fad. This made it difficult to convince them of the value of this thesis project.

Those that were totally unfamiliar with systems thinking or system dynamics were open to the idea, but the approach had to be explained to them. This was fine, except when members of the team changed due to the reorganization or other issues, and explanations had to be repeated and new members brought up to speed. It is very important to have a consistent team for the course of the project in order to progress in a timely manner.

Future Work

Overall, ProductCo has benefited from this project. No concrete policies have been developed, but enough interest has been generated to ensure the continuation of the effort. The insights found so far have aided the members of the team in thinking in a more systemic way about the decisions that they make and the repercussions for the company.

[8] Senge, P. M. 1990. *The Fifth Discipline: The Art and Practice of the Learning Organization.* New York: Doubleday.

It is anticipated that this effect can be enhanced by working with the new Program Managers on the continuing efforts. By the nature of their job descriptions, they have responsibilities that start at the quote response and end with delivery of the final project. If anyone can apply systemic insights and implement policies that impact the whole process, it will be them.

Therefore, it is suggested that ProductCo continue with the modeling effort. This would require reconvening the team, perhaps adding some members from the new Program Management structure. The current structure of the model could be reviewed and validated in light of the new product development process.

Additional causal loops could then be proposed and added to the model to increase its correlation to the real world system. This would help ProductCo to gain further understanding of what is going on with their profitability.

Finally, policies could be proposed to improve the situation. These could be tested in the virtual world by modeling them onto the completed structure. This would give ProductCo insight into the potential consequences of their policy decisions before they are made in the real world.

In summary, system dynamics can and has helped ProductCo improve its product development process. These benefits will be enhanced if the effort is continued.

References

Forrester, J. W. 1961. *Industrial Dynamics.* Portland, OR: Productivity Press.

Ford, D. N. and J. D. Sterman. 1998. Dynamic modeling of product development processes. *System Dynamics Review* 14(1), 31 - 68.

Hines, J. H. 1999. Class Notes for 15.876, Fall 1999.

Hines, J. H. 1999. Class Notes for 15.980, Summer 1999.

Hines, J. H. 1998. Class Notes for 15.982, Summer 1998.

Kreutzer, D. P. and V. Wiley. Undated. Making It Happen: The Implementation Challenge. http://www.gka.com/papers/implement.htm

Lyneis, J. M. 1998. Class Notes for 1.432, Fall 1998.

Lyneis, J. M. 1999. System dynamics for business strategy: a phased approach. *System Dynamics Review* 15(1), 37 - 70.

Morecroft, J. D. W. and J. H. Hines. 1985. Strategy and the Design of Structure. Presented at the 1985 Strategic Management Society Conference, Barcelona, Spain.

Senge, P. M. 1990. *The Fifth Discipline: The Art and Practice of the Learning Organization.* New York: Doubleday.

Sobek, D. K., J. K. Liker, and A. C. Ward. 1998. Another Look at How Toyota Integrates Product Development. *Harvard Business Review*, July - August 1998.

Sterman, J. D. 1998. *Business Dynamics: Systems Thinking and Modeling for a Complex World.* Partial Draft, Version 1.0, May 1998.

Appendix A: Model Documentation

Model for Hypothesis One

This section contains the equations used in the model for Hypothesis One, titled Pressure to Increase Margins. It contains three groups. The group "project" covers equations that deal with the project being bid, won, lost, and completed. The group "return" contains the equations that calculate returns, as well as associated variables. There is also a "control" group which contains the parameters which control the simulations.

Each equation is followed by the units of the variable being calculated, and a short description of the meaning of the variable.

The sketches associated with the model can be found in the text of the thesis.

```
********************************************************
        .project
********************************************************~
A projects available= INTEG (bidding on A projects-(winning A projects+losing A projects),10)
        ~       projects
        ~       Stock of A projects that have been bid on but have not been decided.

losing A projects=(A projects available*(1-percent of A projects won))/"time to win/lose project"
        ~       projects/Month
        ~       Flow of A projects that are not won.

B projects available= INTEG (bidding on B projects-(winning B projects+losing B projects),10)
        ~       projects
        ~       Stock of B projects that have been bid on but have not been decided.

losing B projects=(B projects available*(1-percent of B projects won))/"time to win/lose project"
        ~       projects/Month
        ~       Flow of B projects that are not won.
```

A projects obsoleting=Active A projects/time to obsolete

~ projects/Month

~ A projects predicted to leave the system based on the normal time to \
complete.

B projects obsoleting=Active B projects/time to obsolete

~ projects/Month

~ B projects predicted to leave the system based on the normal time to \
complete.

bidding on A projects=5

~ projects/Month

~ Number of A projects that the company bids on per month.

bidding on B projects=10

~ projects/Month

~ Number of B projects that the company bids on per month.

Active A projects= INTEG (+winning A projects-A projects obsoleting,10)

~ projects

~ Stock of A projects that are currently being worked on.

Active B projects= INTEG (+winning B projects-B projects obsoleting,10)

~ projects

~ Stock of B projects that are currently being worked on.

percent of A projects won=0.6

~ fraction

~ Percent of A projects that are won.

percent of B projects won=0.25

~ fraction

~ Percent of B projects that are won.

time to obsolete=36

~ Month

~ Time required for an active project to be completed and leave the system.

"time to win/lose project"=2

~ Month

~ Average time required by the customer to determine if a quote is accepted.

total active projects=Active A projects+Active B projects
~ projects
~ Total number of active A and B projects in the system.

winning A projects=
(A projects available*percent of A projects won)/"time to win/lose project"
~ projects/Month
~ Flow of A projects that are won.

winning B projects=(B projects available*percent of B projects won)/"time to win/lose project"
~ projects/Month
~ Flow of B projects that are won.

****
.return
**~

allocation percent for A=0.5
~ fraction
~ Percentage of the gap in average return that the company will make up by \
changing the return on A projects.

allocation percent for B=0.5
~ fraction
~ Percentage of the gap in average return that the company will make up by \
changing the return on B projects.

Gap allocated to B=allocation percent for B*Gap in average
~ fraction
~ Actual gap to be made up by changing the return on B projects.

Gap allocated to A=Gap in average*allocation percent for A
~ fraction
~ Actual gap to be made up by changing the return on A projects.

Return Applied to A projects=Price Pressure on A*Indicated Return on A projects
~ fraction
~ Return actually applied when determining price of A.

Return Applied to B Projects=Price Pressure on B*Indicated Return on B projects
~ fraction
~ Return actually applied when determining price of B.

Price of B project=Cost of B project*(1+Return Applied to B Projects)

~ dollars/project
~ Price quoted to the customer for a B project.

Price of A project=Cost of A project*(1+Return Applied to A projects)
~ dollars/project
~ Price quoted to the customer for an A project.

Relative Price of B project=
Price of B project/(Competition's Price for B+Price of B project)
~ fraction
~ Difference between our price and our competition's price.

Indicated Return on B projects= INTEG (Change in return on B projects,0.05)
~ fraction
~ Return that the company would like to apply to B projects in order to \
reach its desired average return.

Competition's Price for A=102
~ dollars/project
~ Price that the competition is charging for an A project.

Competition's Price for B=104
~ dollars/project
~ Price that the competition is charging for a B project.

Time to Change Perception=3
~ Month
~ Time to change perception, based on time between price comparisons \
(quarterly).

Change in Customer Perception of A=Perception Gap A/Time to Change Perception
~ fraction/Month
~ Changes in the customer's perception of the price gap in response to \
changes in the actual gap.

Customer Perception of Relative Price of A= INTEG (Change in Customer Perception of A,0.5)
~ fraction
~ The customer's perception of the company's price for A relative to \
competition. Changes more slowly than the actual gap due to periodic price \
comparisons (not continuous).

Customer Perception of Relative Price of B= INTEG (Change in Customer Perception of B,0.5)
~ fraction

~ The customer's perception of the company's price for B relative to \
competition. Changes more slowly than the actual gap due to periodic price \
comparisons (not continuous).

Perception Gap A=Relative Price of A project-Customer Perception of Relative Price of A
~ fraction
~ Gap between customer's perception of gap and actual gap.

Perception Gap B=Relative Price of B project-Customer Perception of Relative Price of B
~ fraction
~ Gap between customer's perception of gap and actual gap.

Price Pressure on A=Price Pressure f(Customer Perception of Relative Price of A)
~ dmnl
~ Translates "customer perception of relative price of A" into an effect on \
the applied return through the table function "price pressure f"

Change in Customer Perception of B=Perception Gap B/Time to Change Perception
~ fraction/Month
~ Changes in the customer's perception of the price gap in response to \
changes in the actual gap.

Relative Price of A project=Price of A project/(Competition's Price for A+Price of A project)
~ fraction
~ Difference between our price and our competition's price.

Price Pressure on B=Price Pressure f(Customer Perception of Relative Price of B)
~ dmnl
~ Translates "customer perception of relative price of B" into an effect on \
the applied return through the table function "price pressure f"

Price Pressure f([(0,0)-(1,1)],(0,1),(0.5,1),(0.515,0),(0.9,0),(1,0))
~ dmnl
~ Table function.

Total cost of B projects=Cost of B project*Active B projects
~ dollars
~ Total cost of all active B projects.

Change in return on A projects=Gap allocated to A/time to adjust return
~ fraction/Month
~ Change in the % return that the company would like to apply to A projects.

Change in return on B projects=Gap allocated to B/time to adjust return
~ fraction/Month
~ Change in the % return that the company would like to apply to B projects.

Total cost=Total cost of A projects+Total cost of B projects
~ dollars
~ Total cost of all active projects.

Average return=(Total revenue/Total cost)-1
~ fraction
~ % return on sales of the average project.

Total revenue=Total revenue from A projects+Total revenue from B projects
~ dollars
~ Total revenue (price) of all active projects.

Indicated Return on A projects= INTEG (Change in return on A projects,0.02)
~ fraction
~ Return that the company would like to apply to A projects in order to \
reach its desired average return.

Total cost of A projects=Cost of A project*Active A projects
~ dollars
~ Total cost of all active A projects.

Total revenue from A projects=Price of A project*Active A projects
~ dollars
~ Total revenue (price) of all active A projects.

Total revenue from B projects=Active B projects*Price of B project
~ dollars
~ Total revenue (price) from all active B projects.

Cost of B project=100
~ dollars/project
~ Project cost used to develop the price of a B project.

Cost of A project=100
~ dollars/project
~ Project cost used to develop the price of an A project.

Gap in average=Desired average return-Average return
~ fraction
~ Gap between the desired average return and the actual calculated average \

return.

Initial Target A=0.05
~ fraction
~ Intial value for indicated return on A.

Desired average return=0.08
~ fraction
~ The goal for the average return on a project.

Initial Target B=0.1
~ fraction
~ Intial value for indicated return on B.

time to adjust return=12
~ Month
~ Time required for the company to change its return level used for bidding.

****
.Control
**~
Simulation Control Paramaters

FINAL TIME = 300
~ Month
~ The final time for the simulation.

INITIAL TIME = 0
~ Month
~ The initial time for the simulation.

SAVEPER = TIME STEP
~ Month
~ The frequency with which output is stored.

TIME STEP = 0.0625
~ Month
~ The time step for the simulation.

Model for Hypothesis Two

This section contains the equations used in the model for Hypothesis Two, titled Cost Cutting Through Resource Reduction. It contains seven groups. The group "resource" covers equations for resource levels and how they are calculated. The "pdy" group is associated with productivity and all related variables. The "cost" group contains items which deal with actual and bid costs. The group "work" contains the equations that govern work and how it moves through the system. The "quality" group covers the calculations associated with the quality of work being accomplished. The "init" group initializes the model to ensure that the number of tasks in the system agrees with the number of initial projects. Again, there is also a "control" group which contains the parameters which control the simulations.

Each equation is followed by the units of the variable being calculated, and a short description of the meaning of the variable.

As with Hypothesis One, the sketches associated with the model can be found in the text of the thesis.

```
********************************************************
        .init
********************************************************~
Allocation to correct work=0.45
        ~       fraction
        ~       Fraction of the total initial tasks that are in the "correct work" stock.

Allocation to undiscovered rework=1-(Allocation to correct work+Allocation to work to do)
        ~       fraction
        ~       Fraction of the total initial tasks that are in the "undiscovered rework" \
                stock.

Allocation to work to do=0.45
        ~       fraction
        ~       Fraction of the total initial tasks that are in the "work to do" stock.
```

Initial value for correct work=Total initial tasks*Allocation to correct work
~ tasks
~ Initial number of tasks in the stock "correct work".

Initial value for undiscovered rework=Total initial tasks*Allocation to undiscovered rework
~ tasks
~ Initial number of tasks in the stock "undiscovered rework".

Initial value for work to do=Total initial tasks*Allocation to work to do
~ tasks
~ Initial number of tasks in the stock "work to do".

Total initial tasks=Initial Active Projects*Tasks per project
~ tasks
~ Total number of tasks for the initial value of active projects. Must be \
distributed among "work to do", "correct work", and "undiscovered rework".

****
.resource
**~

Perceived minimum resources=Perceived minimum resources per project*winning projects
~ people
~ Perceived minimum resources required to complete the existing projects.

Perceived minimum resources per project=100
~ people/(project/Month)
~ Minimum people required to complete one project won per month.

Effect of Work to Do on Desired Resources f([(0,0)-
(1,20)],(0,1),(0.5,1),(0.649547,3.85965),(0.873112,10.9649),(1,20))
~ dmnl
~ Table function.

Desired resources=Resources*Effect of Pressure on Desired Resources*Effect of Work to Do on Desired Resources
~ people
~ Goal for the level of resources. Effects are due to pressure to reduce \
costs and amount of work to be done.

Adding and removing resources=(Resource gap/Time to adjust resources)
~ people/Month
~ The flow of people in and out of the resource pool in response to the \
desired level.

Effect of Work to Do on Desired Resources=Effect of Work to Do on Desired Resources f(Relative work to do)

~ dmnl

~ Translates "relative work to do" into an effect on the desired resources \
through the table function "effect of work to do on desired resources f"

Cost per resource=30

~ dollars/person

~ Cost that the company bears for each resource.

Effect of Pressure on desired f([(-1,0)-(1,1)],(-1,1),(0,1),(0.154079,0.254386),(0.468278,0.0964912),(1,0))

~ dmnl

~ Table function.\!\!

Effect of Pressure on Desired Resources=Effect of Pressure on desired f(Pressure to reduce costs)

~ dmnl

~ Translates "pressure to reduce costs" into an effect on the desired \
resources through the table function "effect of pressure on desired f"

Initial resources=500

~ people

~ Initial resources assigned to the project.

Resource gap=Desired resources-Resources

~ people

~ Gap between current level of resources and desired level of resources.

Resources= INTEG (Adding and removing resources,Initial resources)

~ people

~ Current level of resources.

Resources relative to min=Resources/(Perceived minimum resources+Resources)

~ fraction

~ A measure of how close the resource level is to the perceived minimum \
required to do the job. This impacts quality.

Time to adjust resources=6

~ Month

~ Indicates how long it takes to hire or fire people.

****

.pdy
***~

Effect of Resource Shortage on PDY=Effect of Resource Shortage on PDY f(Resources relative to min)

~ dmnl

~ Translates the resource shortage into an effect on productivity through \
the table function "effect of resource shortage on PDY f"

Productivity=Normal PDY*Effect of lack of work on PDY*Effect of Resource Shortage on PDY

~ tasks/(Month*person)

~ Current value of productivity. This drops off if there is no work \
available to be done. It also can be affected by the relative resource \
level.

Effect of Resource Shortage on PDY f([(0,0)-(1,1.5)],(0,1.2),(0.25,1.2),(0.501511,1.17105),(0.75,1.1),(1,1))

~ dmnl

~ Table function.

Work to Do required to keep busy=Normal PDY*Resources*Time Horizon for busyness

~ tasks

~ Based on the number of resources, calculates the level of work to do that \
will keep them busy for the duration of the time horizon.

Time Horizon for busyness=6

~ Month

~ Used to determine amount of work to do required to keep the resources \
busy. This is the duration for which we want to keep them busy.

Normal PDY=20

~ tasks/(Month*person)

~ Normal level of productivity, acted on by the effects to determine the \
current value of productivity.

Effect of lack of work on PDY f([(0,0)-(1,1)],(0,0),(0.151057,0.157895),(0.29003,0.815789),(0.5,1),(1,1))

~ dmnl

~ Table function.

Effect of lack of work on PDY=Effect of lack of work on PDY f(WorkToDo/(Work to Do required to keep busy+WorkToDo))

~ dmnl

~ Translates the level of work to do into an effect on productivity through \

the table function "effect of lack of work on PDY f"

****

.cost

**~

Bid Cost=10000

~ dollars

~ Project cost used to develop the price of a project.

Actual Cost=Cost due to Materials+Cost Due to Engineering Resources

~ dollars

~ The current value for the actual cost of the project, based on resource \
level.

Cost due to Materials=0

~ dollars

~ A place to add future model structure for cost of material used in a \
product, which is also affected when there is pressure to reduce costs.

Cost Due to Engineering Resources= Resources*Cost per resource

~ dollars

~ Total actual cost that is due to engineering resources.

Cost Overrun=Actual Cost-Bid Cost

~ dollars

~ Difference between cost used for bidding and actual cost of project.

Pressure to reduce costs=Cost Overrun/(Bid Cost+Cost Overrun)

~ fraction

~ Measure of the extent of cost overrun. Close to zero is negligible \
overrun, 0.5 means that the cost has reached 2x the bid cost.

****

.work

**~

winning projects=3+(step in projects*switch for step)

~ projects/Month

~ Projects that are won by the company per month.

step in projects=STEP(1, 500)

~ dmnl

~ Step function used to kick the system out of equilibrium.

switch for step=0
~ dmnl
~ Switch to turn the step function on or off. 0 = off, 1 = on.

Initial Active Projects=125
~ projects
~ Initial value for the number of active projects.

Fractional Undiscovered Rework=UndiscoveredRework/(CorrectWork+UndiscoveredRework)
~ fraction
~ Amount of undiscovered rework as a fraction of the total work that is \
perceived as accomplished.

Tasks Leaving System 2=Actual completion rate*Fractional Undiscovered Rework*Tasks per project
~ tasks/Month
~ Tasks exiting the system as projects are completed. Some exit from \
correct work, others from undiscovered rework.

Fractional Correct Work=CorrectWork/(CorrectWork+UndiscoveredRework)
~ fraction
~ Amount of correct work as a fraction of the total work that is perceived \
as accomplished.

Tasks Leaving System 1=Actual completion rate*Fractional Correct Work*Tasks per project
~ tasks/Month
~ Tasks exiting the system as projects are completed. Some exit from \
correct work, others from undiscovered rework.

Relative rework discovery=DiscoveringRework/(DiscoveringRework+Tasks entering system)
~ fraction
~ The fractional amount of rework included in the total inflow to Work to Do.

Average time to complete project=36
~ Month
~ Average time required to complete a project.

Effect of rework discovery on completion=Effect of rework discovery on completion f(Relative rework discovery)
~ dmnl
~ Translates "relative rework discovery" into an effect on the completion \
rate through the table function "relative rework discovery f". Reduces \
the rate of completion if too much rework is included in the inflow to \

Work to Do.

Effect of rework discovery on completion f([(0,0)-(1,1)],(0,1),(0.25,1),(0.5,0.3),(0.8,0.1),(1,0))
~ dmnl
~ Table function.\!\!

Active Projects= INTEG (+winning projects-completing projects,
Initial Active Projects)
~ projects
~ Current number of active projects that are in the system.

Actual completion rate=Desired Completion Rate*Effect of rework discovery on completion
~ projects/Month
~ Rate at which the company is actually completing projects.

completing projects=Actual completion rate
~ projects/Month
~ Number of projects that are completed per month.

Desired Completion Rate=Active Projects/Average time to complete project
~ projects/Month
~ Rate at which the company plans to complete projects.

UndiscoveredRework= INTEG (AccomplishingIncorrectly - (DiscoveringRework +Tasks Leaving System 2),Initial value for undiscovered rework)
~ tasks
~ Stock of work that has been done wrong but has not been discovered yet.

Tasks entering system=Tasks per project*winning projects
~ tasks/Month
~ New tasks entering the system from active projects.

Relative work to do=WorkToDo/(Desired work to do+WorkToDo)
~ fraction
~ Measure of the current level of work to do relative to a desired level. \
This indicates when work is piling up and more resources may be needed.

Desired work to do=100000
~ tasks
~ Ideal level of work to do that keeps projects moving through the system.

AccomplishingCorrectly= AccomplishingWork * Quality
~ tasks/Month

~ Work that is being done right. This is determined by the flow of work \
being done and the current quality level.

AccomplishingIncorrectly = AccomplishingWork * (1 - Quality)

~ tasks/Month

~ Work that is being done wrong. This is determined by the flow of work \
being done and the current quality level.

AccomplishingWork=Resources*Productivity

~ tasks/Month

~ Work getting done. This is dependent on the resources available and their \
productivity.

DiscoveringRework=UndiscoveredRework/Time to discover rework

~ tasks/Month

~ Incorrect work being discovered. This flows back into the pile of work to \
do.

CorrectWork= INTEG (AccomplishingCorrectly-Tasks Leaving System 1, Initial value for correct work)

~ tasks

~ Stock of work that has been done correctly.

Time to discover rework=1

~ Month

~ Time required to discover work that has been done wrong, and send it back \
to be done again.

WorkToDo= INTEG ((Tasks entering system + DiscoveringRework) - AccomplishingWork, Initial value for work to do)

~ tasks

~ Current number of tasks waiting to be accomplished.

Tasks per project=1800

~ tasks/project

~ Number of projects entering the system for each active project.

****

.quality

***~

Quality=Maximum Quality*Effect of Resource Shortage on Quality

~ fraction

~ This fraction determines the split between work done right and work done \

wrong.

Effect of Resource Shortage on Quality=Effect of Resource Shortage on Quality f(Resources relative to min)

~ dmnl

~ Translates "resources relative to min" into an effect on quality through \
the table function "effect of min on quality f". This reduces the quality \
level when the resources are close to their minimum level.

Maximum Quality=0.85

~ fraction

~ Normal value for quality.

Effect of Resource Shortage on Quality f([(0,0)-(1,2)],(0,0),(0.5,0.7),(0.7,0.9),(0.9,1),(1,1))

~ dmnl

~ Table function.\!\!\!

****

.Control

**~

Simulation Control Paramaters

FINAL TIME = 200

~ Month

~ The final time for the simulation.

INITIAL TIME = 0

~ Month

~ The initial time for the simulation.

SAVEPER = TIME STEP

~ Month

~ The frequency with which output is stored.

TIME STEP = 0.0625

~ Month

~ The time step for the simulation.