Is Specialization the Most Effective Solution to Overload?

A System Dynamics Exploration of a Product Development Organization

by

Linda Thi Nguyen

S.B. Mechanical Engineering
Massachusetts Institute of Technology, 1998

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 2010

Signature of Author____
Linda Thi Nguyen
System Design and Management Program
February 2010

Certified by____
J. Bradley Morrison
Thesis Supervisor
Senior Lecturer
MIT Engineering Systems Division

Certified by____
Patrick Hale
Director
System Design & Management Program

Is Specialization the Most Effective Solution to Overload?
A System Dynamics Exploration of a Product Development Organization

by

Linda Thi Nguyen

Submitted to the System Design and Management Program on January 15, 2010
in Partial Fulfillment of the Requirements for the Degree of

Master of Science in Engineering and Management

Abstract

This work investigates the overload problems of one product development department in a consumer products company. Many in the organization attribute the problem to an external source – a burgeoning product portfolio. The most common solution posed is to split the department into two in order to reap the productivity gains of specialization and reduce the overload.

It is proposed that some of the overload is internally generated and specialization alone will not be enough. There is a Reinventing the Wheel phenomenon that occurs when projects are handed off from one person to another. Additional work is generated with each handoff. This problem can be exacerbated by high turnover, lack of documentation, switchloss, or delays in project completion.

System dynamics models were created to explore the feedback loops, delayed effects of managerial decisions and resulting behavior of the system.

Results showed that specialization leads to initially high productivity, but the gains decrease over time as breadth of experience across the organization decays and rework and coordination costs increase. It is also shown that overload could be internally generated through managerial policies. If these policies are not changed, specializing may not be as effective.

Recommendations include considering turnover as part of project planning and carefully monitoring workload so that the productivity does not plummet and affect all programs. Specialization is a good solution in some cases, but is not the best solution for work that requires a great deal of interaction between functions, where the level of coordination required to share knowledge outweighs the productivity gains.

Thesis Supervisor: J. Bradley Morrison
Title: Senior Lecturer of the MIT Engineering Systems Division

(This page intentionally left blank.)

Acknowledgements

This thesis is an ode to Parkinson's Law: work expands to fill the time available.

To my classmates and SDM administrators, I am glad I had the opportunity to meet and work with all of you. I thank you for all the wonderful conversations, meals, late nights, early mornings and best of all, new friendships. I look forward to seeing the SDM community grow into the future.

My family deserves almost as much credit for this thesis as I do. To my sister, we would have starved to death had you not been around to feed us.

To my loving husband and best friend, I could never have made it through the last three years without your unwavering support and encouragement. As I type away late into the hours of darkness, my infant son is sleeping through the night, so for that I am also grateful.

(This page intentionally left blank.)

Table of Contents

List of Figures

List of Tables

1 Introduction

1.1 Motivation

It was important to me to work on a real world problem that will be of value to my company sponsor. The findings of the thesis will have a direct impact on my department, Product Engineering, as well as the product development process of the business.

The Company is a global consumer products business. Product Engineering (PE) resides in the Engineering Group and is responsible for developing the product design and implementing it into production. This is the only group in the company that is involved with a product from concept through implementation and keeps responsibility through long term maintenance as well. This group still supports the first product line launched a hundred years ago. Historically, Product Design and Product Engineering were in one Product Development Organization. As the clockspeed of breakthrough products increased from one every couple decades to one every 10 years, the group was split into two about 15 years ago. The intent was to get product development engineers closer to the other engineering disciplines (machine design, mold design, die design) that turn the product from a concept on the computer into a tangible product and allow the designers to focus on just designing.

Product Engineering had already been stretched, when personnel turnover, process changes, additional responsibilities and more product launches overwhelmed the department in the last few years. The department has been able to meet business objectives "successfully", but at a significant cost to the quality of work, completeness of work (slacking on documentation), and upstream/downstream customer support.

Management thinks the timing is right for another reorganization. Today, as the product portfolio continues to expand with more and more line extensions and platform products in more and more geographies, management believes splitting the Product Engineering group will resolve existing issues. However, I believe the original split of the product development process has actually caused many delayed effects in the system that Product Engineering has been painfully correcting today (for example, costly design flaws that do not surface until late in the development cycle). I believe that dividing the organization further may exacerbate the current problems.

The short term solution has been to hire more people. However, expanding headcount cannot continue indefinitely. It is also straining existing resources bringing the new hires up to speed. It takes about 8 years to become a product "expert" by taking a breakthrough product from design through to launch and getting to see the consequences of early decisions.

The scope of this thesis will be constrained to one department with approximately 20 engineers; however, the intent is to make it scalable to the wider organization in the future. The lesson from Wheelwright and Clark's case study, Creating Project Plans to Focus, is to consolidate the product portfolio and aggregate planning to ensure the organization is not overloaded. However, in a large business, especially given stockholder pressures to increase sales and continue aggressive growth, it is often not realistic for a department manager to be able to influence the greater business in a timely fashion. It is also typical for headcount to be capped, and managers are asked to do more with less. Given minimal influence on the rest of a very large business to reduce the number of products in the portfolio and development pipeline, how can a department be managed to best respond to the needs of the business?

1.2 Hypothesis

Reorganization may be the most apparent solution, but in and of itself may not be sufficient to improve the overall system effectiveness for the whole product development process. It may have longer term unintended consequences that should at the very least be identified, acknowledged, and mitigated.

There are internal practice generates additional work and compounds the effects of the expanding product portfolio. High turnover and constant reshuffling of work leads to knowledge loss with each handoff from person to person, leading to redoing work others may have already done.

I propose that specialization alone is not a sufficient solution because current policies and behaviors elicit Reinventing the Wheel.

1.3 Methodology

This is a case study based on the author's job. The qualitative data was generated from the author's own experience, discussions with managers and colleagues, and literature. A literature search was conducted on several subjects: multitasking, overloaded organizations, system dynamics modeling, knowledge management, and specialization (or division of labor).

System dynamics modeling is best applied to situations that have long delays between cause and effect, as a tool to explore mental models and beliefs in how the system works, where results are not obvious, and where structure is expected to influence behavior (Sterman 2000). Because the situation with Product Engineering meets this description, system dynamics was chosen as the applied technical methodology. Despite best intentions, the department is stuck in a vicious cycle and is looking for the high leverage policies to help regain order. Just looking at a problem through the lens of system dynamics (generating causal loops, describing variables and examining the structure) can yield great insights to the behavior in the system without running any simulations.

After identifying the key issues PE is facing today and the key measures of interest, dynamic hypotheses were generated to describe possible causes. Two models were developed to simulate these hypotheses. Reinventing the Wheel and Specialization. Through an iterative process, the models were refined from the learnings generated from initial simulations. Data from the literature was used to supplement where the author's experience could not be applied.

One of the benefits of these models is to serve as a management simulator to test out different possible future scenarios and examine the effects of different courses of action. The purpose of modeling is to gain insight to management decisions and their effects on system behavior. As with any model, it is not equal to reality. Many simplifications were applied as needed. However, the baseline behavior of the models qualitatively represents what is seen in real life. The purpose of modeling is NOT to do quantitative predictions and should not be used as such.

(This page intentionally left blank.)

2 Case Study Background

2.1 Organizational History

Gamma is a global consumer products company with a long history of manufacturing components on the order of billions. The product development process typically takes a product from concept through worldwide mass production through several ramp-up phases: Design, Development, Implementation, and Manufacturing. Primary manufacturing processes include stamping, molding, and high precision assembly.

The Design phase includes activities such as: technology transfer from R&D, concept generation to best incorporate the new technology, industrial design to get an aesthetically appealing product, creating the computer generated (CAD) model, rapid prototyping, consumer testing to meet performance objectives, and single cavity tooling to create product for testing.

The Development phase takes the concept and makes it manufacturable by increasing the robustness of the design, making it more tolerant to variation. In real life, parts are not as perfect as the computer model. Imperfections from molding of plastic or stamping of metal or die casting leads to variation in components which can lead to variation in assembly and product performance. This variation is introduced when molds are created and the number of cavities increases. During this phase, parts are typically created in 4 cavity molds, which mean 4 times the variation that was seen in single cavity tooling. Product testing is conducted to ensure performance is maintained throughout.

The Implementation phase brings the process from the Development phase up to mass production quantities. Assembly trials, tooling qualifications, and vendor validations, are all required to ensure that the product design intent is maintained as the company ramps up to produce billions of units. Implementation includes working with vendors, design of experiments to optimize processes, validation of gauging and inspection, monitoring processes and failure testing to ensure a robust design, and introducing variation into processes to ensure stability over long life of the product. This phase also includes debugging equipment (assembly, molding, stamping) and understanding the interactions between product and equipment to troubleshoot feeding jams, for example.

The Manufacturing phase takes the product from launch through obsolescence. Since the company rarely discontinues a product, the maintenance phase can last for quite some time. It can include cost reductions, tooling refurbishments as they age, supply interruptions (changing vendors), moving equipment from plant to plant, and modifications to the product in response to consumer complaints or scrap reduction efforts.

Historically, new products have been introduced with decades in between releases. In the past, one product development group took the product from technology transfer through design, development, implementation and manufacturing. This one group owned and maintained product specifications for the life of the product as shown in Figure 1. Typically, one person became the expert for each product, owning it for decades. Since the products rarely become obsolete, product life can last over 100 years.

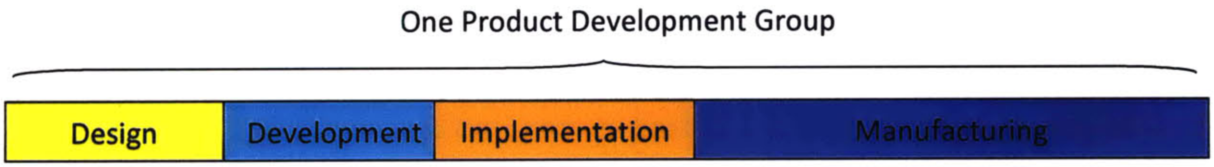


Figure 1: Historical Product Development Process

Over time, marketing objectives required reapplying existing technology to new market segmentations (high/low price points, male/female differentiation, developed/emerging markets). In response to decreasing time to market during the 80's, the product development group was split into two departments, Product Design (PD) and Product Engineering (PE) with responsibilities split as shown in Figure 2. This is a very large company and though many other groups are involved in taking a product to market (machine design, molding, die makers, marketing, sales, planning, finance, and manufacturing to name a few), the scope of this thesis is to focus on the part of the product development process that involves these two groups because there is a handoff of product ownership that occurs between PD and PE once the product performance has been established. PD takes the product from technology transfer to performance establishment by meeting program objectives. The product is then transferred to PE to implement into manufacturing and maintain for the life of the product. Because of this handoff, PE's workload is highly coupled to the quality of PD's work. PE is the main department of interest. Because it owns the product for the rest of its life, the scope of ownership in PE never decreases. This has contributed to the overload of work.

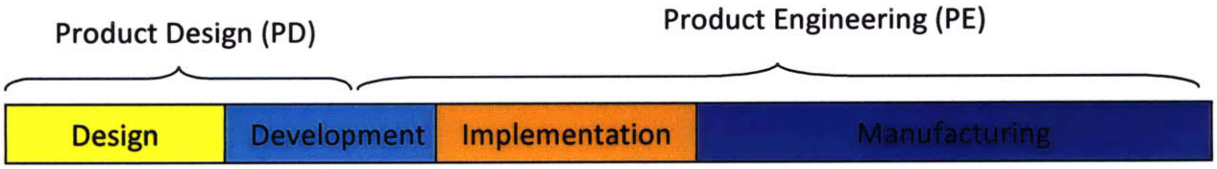


Figure 2: Initial Specialization in the 80's

Today, the level of consumer segmentation continues to increase. Using the terminology and definitions from Wheelwright and Clark (1992), there are three types of new product projects: breakthrough, platform, and derivative. Breakthroughs are next generation products with significant design changes or new manufacturing processes. Platform products are based on the same Breakthrough technology applied to a new segment. Derivatives are minor changes (typically color changes) that are held to the existing specifications and performance criteria. For the purposes of this thesis, I will also add a fourth project category: Maintenance, which covers all activity associated with the product after it is launched into the market. This is relevant to Product Engineering because of the lifetime ownership of the product.

For a typical breakthrough product (next generation) this process from Design, Development, and Implementation can take as long as 10 years. Unlike the consumer electronics industry where products are obsolete within 6 months, the Manufacturing phase can be at least 40 years making the lifecycle of Gamma products 50+ years. During this period, the product can be launched around the world, the manufacturing process can be moved from one plant to another or a brand new line put into a new plant. Over the years, supplies can be interrupted as vendors discontinue materials. Cost reductions are always sought after each year. All of this activity requires engineering time to ensure that product performance and manufacturability is not degraded over time. These are the projects that fall into the Maintenance category. It is the Maintenance work that continuously grows with each additional new product.

2.2 Current Product Development Process

Historically, there were clear cut lines between each phase. However, the pursuit of faster speed to market increased the level of concurrent engineering, i.e. as timelines were condensed, the different phases began overlapping each other (Figure 3).

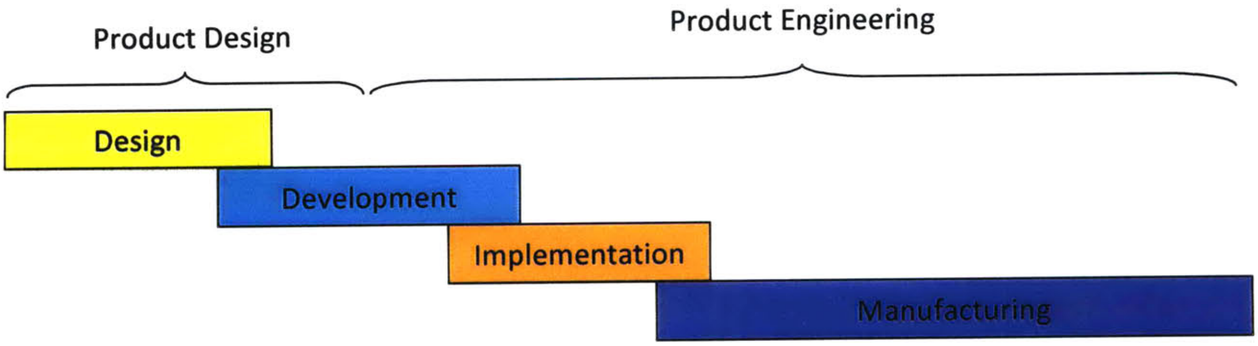


Figure 3: Current Product Development Process

Today, as a consequence of the timeline compression, machine design and mold design cannot wait for a fully vetted product design before starting their work. Due to long lead times, production tools need to be ordered while designs are being tested. In extreme cases, designs are still being tweaked as production is starting up, and design changes are implemented after launch.

New Product Type	Time Between Launches
Breakthrough	8 years
Platform	2 years
Derivative	6 months – 1 year

Table 1: New Product Types and Cycle

Table 1 summarizes the current launch cycles. Breakthrough programs went from every 50 to 30 to 20 to now 8 years. Platform programs now occur every 2 years, and Derivatives are every 6 months. Compounding this speedup effect is further segmentation of high and low cost products for established and emerging markets which means multiple platform product launches occurring simultaneously. Additionally, each new product gets a new color every year.

The PE department has not changed to adapt to the proliferation of programs. It is recognized that the current organizational architecture is not sustainable given the current product portfolio. As depicted in Figure 4, conventional wisdom says that the product development process should now be split into 3 groups: Product Design, Product Engineering, and Product Maintenance.

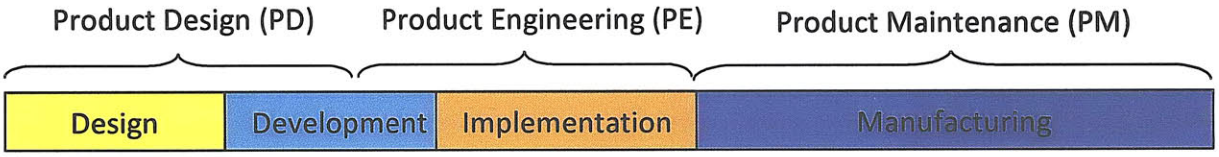


Figure 4: Proposed Specialization

As shown in Table 2: Phases associated with each project type, the process can be broken up by Project Type and Phase.

Project Type	Design Phase	Development Phase	Implementation Phase	Manufacturing Phase
Breakthrough	X	X	X	X
Platform	X	X	X	X
Derivative	N/A	N/A	X	X
Maintenance	N/A	N/A	X	X

Table 2: Phases associated with each project type

The groups could potentially be assigned per Table 3.

Project Type	Design Phase	Development Phase	Implementation Phase	Manufacturing Phase
Breakthrough	PD	PD/PE	PE	PM
Platform	PD	PD/PE	PE	PM
Derivative	N/A	N/A	PE	PM
Maintenance	N/A	N/A	PM	PM

Table 3: Potential Breakdown of Group Responsibilities

Ideally, all new product programs should be staffed with Product Engineers early in the design phase so that Product Design (PD) transfers the design intent and knowledge of key consumer performance characteristics to Product Engineering (PE) and PE transfers manufacturing knowledge and learnings from the consumer to PD. However, in reality, there are never enough resources to staff programs early enough. Resources are too overloaded with implementation and manufacturing issues of today to worry about potential problems of tomorrow, so they do not get involved in early design reviews and feasibility studies. By the time PE does get involved with a new product program, the design is already set and there is no time to change anything but the most critical. This is a recognized problem in the department as well as industry (Repenning 2000) and will not be directly addressed in this thesis. The hope is that insights from this thesis will help alleviate some of the downstream overload issues to free up resources who can then be assigned to new product programs much earlier.

2.3 Problem Definition

2.3.1 Key Issues

The following complaints are based on the author's personal experience, discussion with the current department manager, and discussions with other team leaders in the department.

1. Programs are never 100% complete. The documentation never gets done. It only gets so far and then it takes years to complete.
   - For example, it took 30 years to complete a production release for one of the product families.
2. Headcount keeps increasing, but it cannot increase forever.
   - The department has grown by 20% over the last 5 years in response to the increased workload, but has hit the maximum.

3. The assigned PE engineer changes with the wind.

- A colleague in another department that typically does not have high turnover (experts with 20+ years in the same position is common) once complained that he has had to teach 20 new product engineers about the same manufacturing process. Every year someone new gets assigned and he has to take the time to get them up to speed.

4. This is like the 5th time I had to solve this same problem!
5. Workload never decreases.
6. Projects take too long to get going.
7. Designers are not held responsible for the problems they send downstream.
8. I'd have more time to do real work if we didn't have so many meetings!

2.3.2 Root Cause Hypotheses

The Key Issues Identified above were then attributed to possible explanations:

Rework Cycle

1. Programs are never 100% complete

Reinventing the Wheel

2. Headcount keeps increasing, but it cannot increase forever.
3. Assigned PE engineer changes with the wind.
4. This is like the 5th time I had to solve this same problem!

Overload and Multitasking

5. Workload never decreases.
6. Projects take too long to get going.

Specialization

7. Designers are not held responsible for the problems they send downstream.
8. I'd have more time to do real work if we didn't have so many meetings!

These themes were tied together through causal loops which became the basis for the modeling work. They will be discussed further in subsequent chapters.

2.4 Reference Modes

Based on the problems above, key measures of interest are listed below. Reference modes depict the dynamic behavior of the key measures over a period of time. Each reference mode below shows the historical, present, and possible future behavior of each variable.

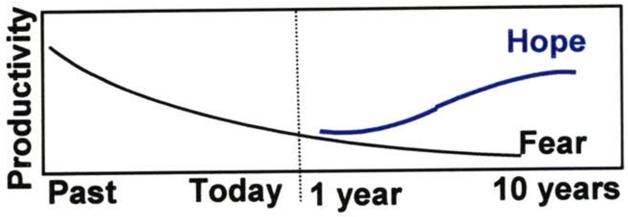
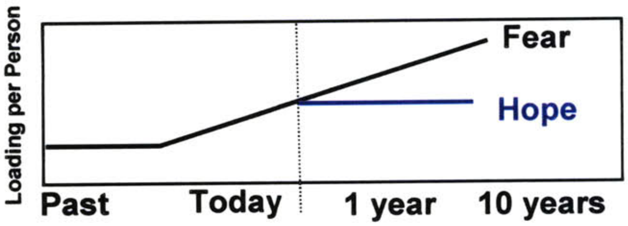
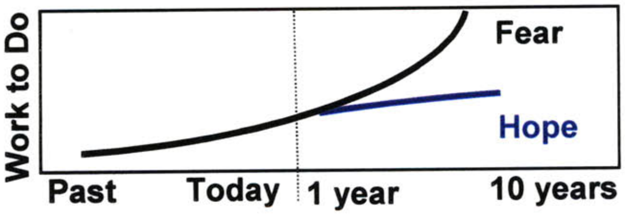
	The fear is that productivity will continue to decline. The hope is that the reorganization will increase productivity.
	Historically, loading was steady. Then, the portfolio kept expanding which increased work, but no new headcount was allowed. The hope is that the productivity improvements will stabilize the workload.
	The To Do list of work keeps growing. Low priority projects just never seem to go away. The hope is that the list will at least become stable.

Figure 5: Reference Modes

2.5 System Boundary/Problem Scope

The models only simulate the decisions and influence of one department manager. Exogenous variables are out of the manager's control. For instance, the input of new product initiatives is determined by business factors such as corporate growth targets and stockholder pressure. Additionally, max headcount is set by the organization and is often difficult to increase. The manager can only react to the work input and disperse the work among her resources.

(This page intentionally left blank.)

3 Specialization Model

3.1 Basis

> **"The man whose life is spent performing a few simple operations has no occasion to exert his understanding or to exercise his invention and generally becomes as stupid and ignorant as it is possible for a human creature to become."**
> **- Adam Smith, *Wealth of Nations***

Everyone recognizes that PE is an overloaded organization and that something must change. The current situation is unsustainable. The current solution is to divide the group into more specialized departments. The intent is to increase productivity through efficiency gains in order to improve the situation and reduce the overloading. However, there are some unintended consequences to specialization (coordination cost, lack of systems understanding) that everyone should be aware of so that policies could be put in place to mitigate them. As Adam Smith suggests above, if taken to the extreme, specialization could diminish the innovation and problem solving abilities of an organization.

If everyone had unlimited resources and time to do their jobs perfectly, product engineering would not exist. Historically, one of PE's strength had been to fill in the gaps when other departments fall short (in resources or knowledge). For example, design engineers are not as knowledgeable about manufacturing practices and issues downstream. Oftentimes, with pressures to deliver the design, they forget that real world conditions are not as perfect as the CAD model. PE's role during the early phases of the design is to defend manufacturability and remind the designer of past issues and learnings from existing products. PE brings knowledge of variation to improve the design. One example would be pushing for more realistic tolerances on components. It is this push/pull tension between performance and cost/manufacturability that yields the "optimized" product design. At the other end of the process, at times, PE's role in the Manufacturing phase is to play the Quality Engineer role to ensure that original design intent is upheld.

Literature surrounding specialization typically focuses on manufacturing and economic applications (Greenberg, 2003). However, the same principles can be applied to knowledge workers as well. The purpose of this model is to understand the immediate benefits and to explore the longer term delayed effects of specialization.

For specialization to make sense, Productivity Gains must exceed Productivity Losses. Sources of gains include less switchloss because you are performing the same tasks over and over which means that each person can complete more tasks at once. Also, specialization shortens the learning curve because there is less to learn within each specialty and it becomes easier to become an expert. On the other hand, Marengo and Dosi (2005) show that specialization narrows the solution space which reduces the likelihood of finding optimal solutions for any given problem. Hence, losses come from a lack of system understanding leading to higher coordination costs (more meetings and less time to do "real" work) or more rework because downstream issues were not anticipated.

3.2 Structure

Only key variables are described in detail in the thesis. For all others, see Appendix A.

Work Structure

The work structure is based on the typical rework loop used widely among the System Dynamics community. There is a great deal of literature involving the rework cycle model. It has been used to describe numerous product development projects and processes in several industries. Ford and Sterman (2003) applied it to explain the "90% Syndrome" where a project will get 90% complete as planned but take twice the time to finish the last 10%. This delayed completion is due to the undiscovered errors that is generated in early phases and requires costly rework in later phases. The longer the delay to discover rework, the more costly the fix. They conclude that "Increased concurrency interacts unfavorably with the delays in discovery of rework needs. The greater the overlap, the more work is completed and released before rework requirements can be detected."

As it applies in this case, the Product Designer may put in features that are not manufacturable in large quantities. For instance, the part design may eject from a single cavity prototype tool without any issues, but the design is not 100% transferrable to higher cavitation production tooling due to speed and cooling requirements. If a mold designer is not consulted early on, the designer will need to recreate the model and have more prototype parts made to conduct more testing to verify performance. In worst cases, the issue is not discovered until after Production tooling has been ordered or completed and reworking 64 cavities is extremely costly.

The Rework Structure consists of 3 stocks shown in Figure 6: Work to Do, Work Completed, and Undiscovered Rework. When tasks leave Work to Do, a portion goes to Work Complete and the other goes to Undiscovered Rework. The ratio is determined by Fraction Reworked.

Normally, what is called Fraction Reworked in this model is called Quality. Though the structure and therefore behavior of the model remains the same, I wanted to distinguish between the two because they are influenced by different factors. Quality is typically a function of on the job experience, time pressures and fatigue. This has been covered extensively in the literature (Repenning 2001, Ford and Sterman 2003), so it is not included in this model. In contrast, as defined here, Fraction Reworked is directly driven by a lack of system understanding, or breadth of experience. It represents the inability of the upstream designer to anticipate issues that may arise downstream in manufacturing. This leads him to generate rework (unbeknownst to him) that will need to be fixed months (years) later. Many times, due to the delayed effects of his actions, he has no idea that rework even occurred and in some cases repeats the offense over and over again, depending on the delay in communications from manufacturing folks back to designers. Rework can also be generated when a manufacturing engineer changes the product without understanding why the design features were there in the first place. This could cause consumer complaints and rework to fix.

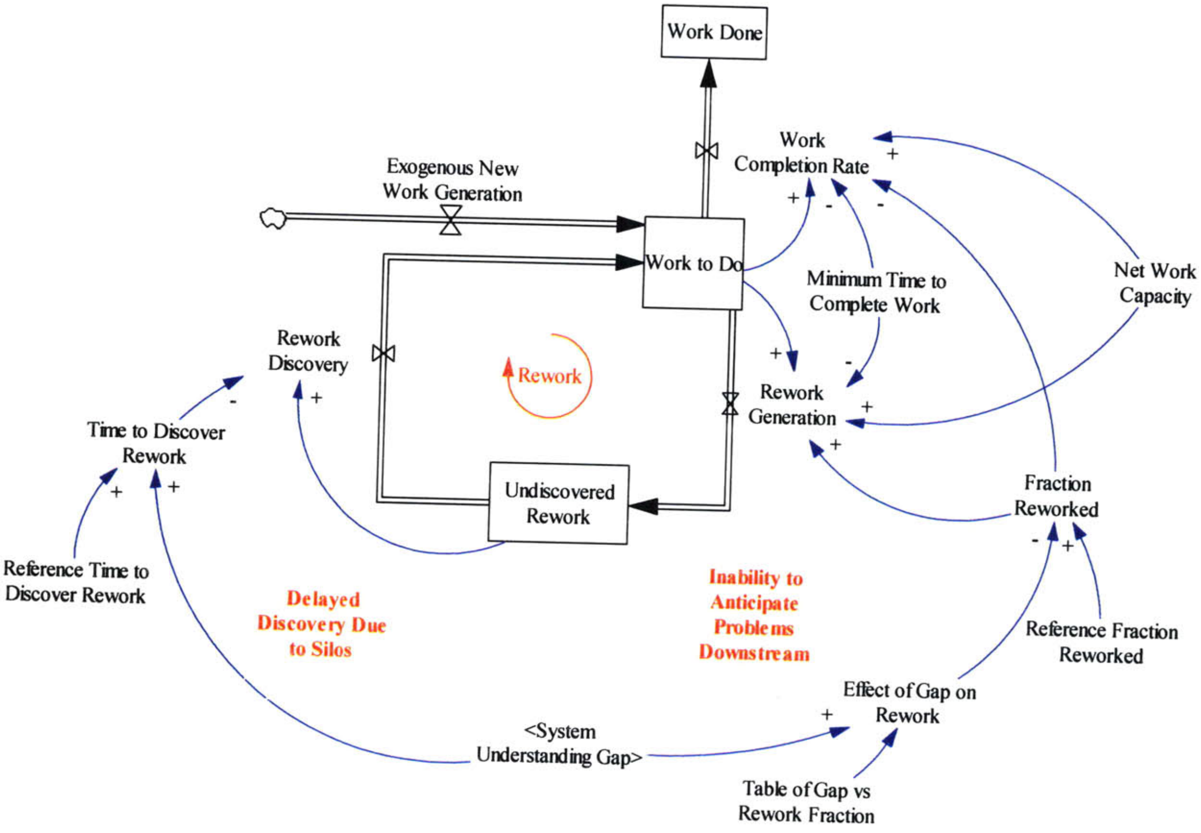

Work Done
Work Completion Rate
Exogenous New Work Generation
Work to Do
Minimum Time to Complete Work
Net Work Capacity
Rework Discovery
Rework
Rework Generation
Time to Discover Rework
Undiscovered Rework
Fraction Reworked
Reference Time to Discover Rework
Delayed Discovery Due to Silos
Inability to Anticipate Problems Downstream
Reference Fraction Reworked
Effect of Gap on Rework
<System Understanding Gap>
Table of Gap vs Rework Fraction
+
-

Figure 6: Specialization Rework Structure

Labor Stock

As Figure 7 shows, Labor is a stock that is increased by Hiring and decreased by Attrition. It is linked to the Experience Structures and drives Increase on the Job Experience and Average Experience.

Experience Structures

There are two structurally identical stock and flows: Depth of Experience and Breadth of Experience (Figure 7). Experience is a stock that accumulates from On the Job Experience and the Experience from Hiring that new hires bring in with them. As people leave through Attrition, they take the Average Experience with them. There is also Experience Decay due to technology changes, process changes, and forgetting by the engineer.

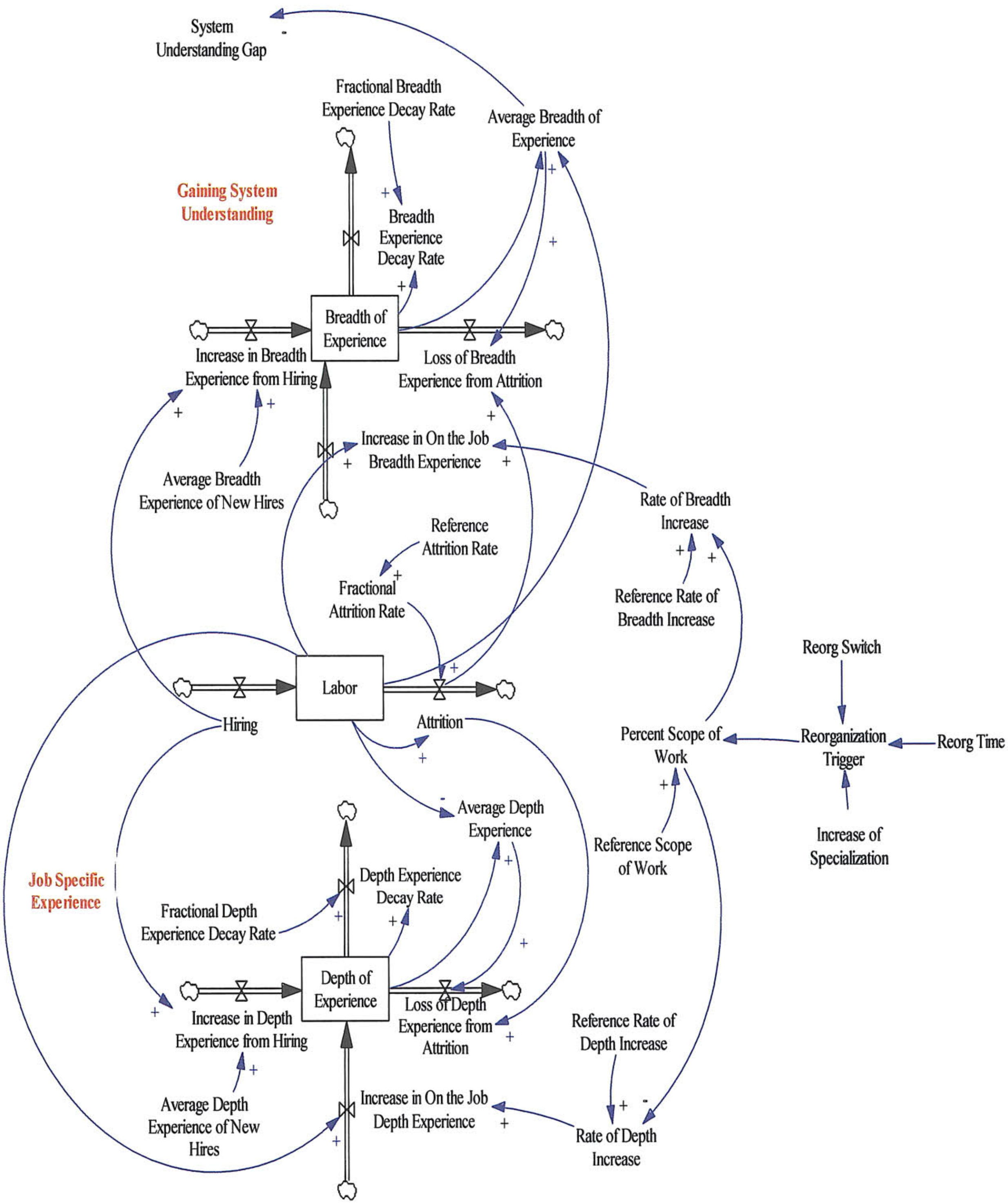

System Understanding Gap
Fractional Breadth Experience Decay Rate
Average Breadth of Experience
Gaining System Understanding
Breadth Experience Decay Rate
Breadth of Experience
Increase in Breadth Experience from Hiring
Loss of Breadth Experience from Attrition
Increase in On the Job Breadth Experience
Average Breadth Experience of New Hires
Rate of Breadth Increase
Reference Attrition Rate
Fractional Attrition Rate
Reference Rate of Breadth Increase
Reorg Switch
Labor
Hiring
Attrition
Percent Scope of Work
Reorganization Trigger
Reorg Time
Average Depth Experience
Reference Scope of Work
Increase of Specialization
Job Specific Experience
Depth Experience Decay Rate
Fractional Depth Experience Decay Rate
Depth of Experience
Increase in Depth Experience from Hiring
Loss of Depth Experience from Attrition
Reference Rate of Depth Increase
Average Depth Experience of New Hires
Increase in On the Job Depth Experience
Rate of Depth Increase

Figure 7: Specialization Labor and Experience Structure

The difference between the two Experience structures is what they represent and the effect that specialization has on them. Depth of Experience represents job specific experience. This has to do with specific tools and processes for your own job. This variable is used to determine productivity in the model. Because specialization lowers the scope of work, it immediately increases the rate of Depth increase, i.e. people learn their job faster if there is less to learn.

On the other hand, Breadth of Experience represents the level of system understanding that each person has. If historically, one person owned the product from design to obsolescence, she would be almost all knowing after one or two times through the process. However, once the work is divided between several departments, the level of system understanding diminishes because people only know their specific portion of the process and it is difficult for knowledge to pass through the departmental barriers, if at all.

Breadth of experience takes a long time to accrue. Less specialization means that everyone gets to work on a broad range of projects and phases. As specialization increases, the accrual rate decreases. Also, since the scope of work is less, new hires do not have to have the same level of experience, so the increase of experience from new hires goes down. Potentially, specialization increases attrition because people get bored faster. The decay rate would increase because people are not keeping up with training for skills that they no longer use. In general, the inflow decreases and the outflow increases due to specialization. For this reason, Breadth of Experience decreases over time.

This indeed happened during the last specialization into two departments. For example, when the product development process was initially split into two, PE started with a high level of CAD (computer aided design) skills and could pitch in to help out with modeling if PD had a shortage of resources which did happen on occasion. However, CAD skills need to be maintained and constantly utilized to stay current with new software. Since PE does not use CAD as much as PD, these skills have decayed over time. Because this is no longer a critical skill for the job, new hires are not required to have any CAD skills. Several of those with a high skill level have since left the department. Today, product engineers only have a rudimentary knowledge of how to use the CAD software. They can make minor drawing changes, but the department has lost the ability to make complex model changes.

Of course, it makes sense that skills that are not needed daily should not be maintained and when such work is needed, a specialist could be hired to do the work on a contract basis. However, the downside is that without the experience of doing design work, it is more difficult to understand the constraints that the designers must face. This understanding facilitates communication and builds working relationships across departmental boundaries. It provides empathy so that PE engineers do not make demands that are deemed unreasonable by the designers. In the case of the proposed specialization, a Product Manufacturing engineer could be so far removed from the design process that he may not fully understand the design intent of a product and make changes/decisions that could be detrimental to the consumer.

Key Variables

Percent Scope of Work
The entire process from Design to Obsolescence would be 100% scope of work. Specialization reduces Scope of Work which reduces the Effect of Switchloss and Rate of Breadth Increase, but increases Rate of Depth Increase. This change occurs instantly when Specialization occurs.

Effect of Switchloss on Productivity
The more work a person has, the greater the losses due to switchtasking. This reduces productivity which reduces the rate of work completion. Specialization reduces the effects of switchlosses because you are doing more of the same. Since the scope of work is reduced, there is not as much relearning something that you have not used in a while. This change occurs instantly when Specialization is triggered.

Rate of Breadth Increase goes down as Percent Scope of Work decreases.

Rate of Depth Increase goes up as Percent Scope of Work decreases.

System Understanding Gap
System Understanding is proportional to Breadth of Experience. An all knowing person would understand the entire process from start to finish.

Coordination Cost increases as System Understanding Gap increases. Because no one person knows the entire process, oftentimes, cross functional teams are needed to communicate across the business. Unfortunately, due to ever increasing specialization in pursuit of productivity gains, corporate America has also seen the proliferation of meetings (MCI Conference, 1998). What used to take one person*hour to complete, now takes an additional hour meeting with 5 people (total of 6 person*hours) to discuss how to accomplish it.

Fraction Reworked
As System Understanding Gap increases, the ability for the designer to anticipate issues goes down which leads to rework when the product gets implemented into production. It also means that the engineer who implements it into production does not fully understand the design intent of the component and may make changes to improve manufacturability that is detrimental to the performance of the product which would also require rework to fix.

Time to Discover Rework
An all knowing person would be able to anticipate issues, probably without realizing it, and the discovery time would be instant. Someone who can anticipate issues will make decisions on a daily basis to avoid problems in the first place. As System Understanding Gap increases, the Time to Discover Rework also increases. This is especially true in this case where the work is passed on from one department to the next. And unfortunately, the designer up front may

never even hear about the problems that manufacturing is having with the design choices that he made 3 years earlier.

Productivity is the number of tasks per unit of effort (person*month). For the purposes of this thesis, productivity pertains to the completion of value added tasks because projects are judged by the completion of concrete deliverables. This means that mentoring, communicating, and attending meetings is not considered productive. Reduced productivity means it takes more effort to complete the same task. It increases with Depth of Experience.

Net Productivity is the result of Productivity decreased by Effect of Switchloss and Effect of Coordination Cost.

Feedback Loops

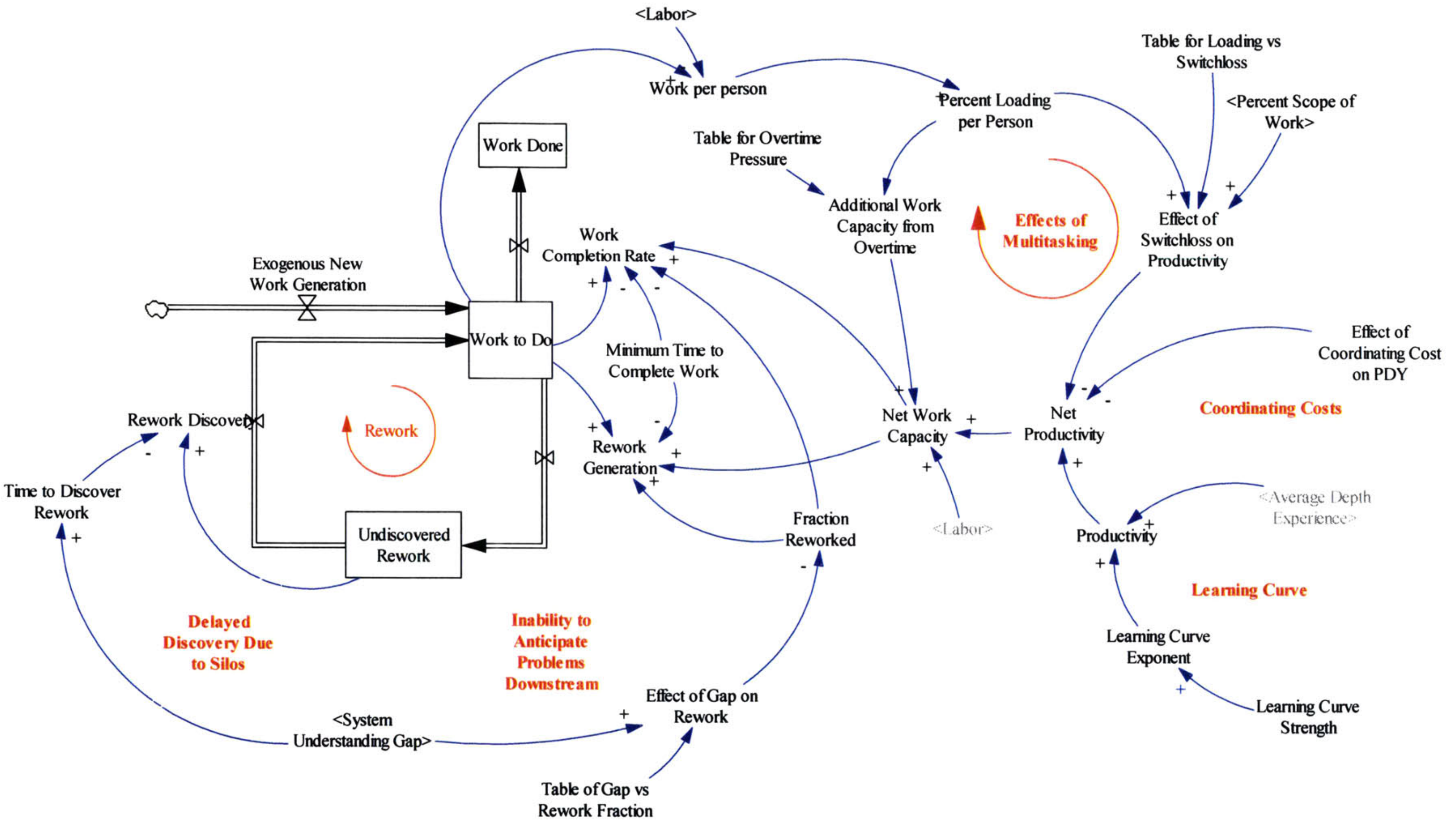
There are two main feedback loops in this model: the Rework Loop and the Multitasking Loop.

Rework

Refer to Figure 8. Starting with Work to Do, as it increases, more rework is generated and sitting in Undiscovered Rework. Once it is discovered, it increases Work to Do. This is a reinforcing loop. As long as there is Work to Do, there is Undiscovered Rework. As a result, Work to Do is never complete.

Multitasking

Starting with Work to Do, as it increases, Work per Person increases, Percent Loading per Person increases, Effect of Switchloss increases, Net Productivity decreases, Work Capacity decreases, Work Completion Rate decreases, and Work to Do increases. This means that it is a reinforcing loop; once it is triggered, it is a downward spiral that keeps piling up the work.



<Labor>
Work per person
Percent Loading per Person
Table for Loading vs Switchloss
<Percent Scope of Work>
Work Done
Table for Overtime Pressure
Additional Work Capacity from Overtime
Effects of Multitasking
Effect of Switchloss on Productivity
Work Completion Rate
Exogenous New Work Generation
Work to Do
Minimum Time to Complete Work
Net Work Capacity
Net Productivity
Effect of Coordinating Cost on PDY
Coordinating Costs
Rework Discovery
Rework
Rework Generation
Time to Discover Rework
Undiscovered Rework
Fraction Reworked
<Labor>
Productivity
<Average Depth Experience>
Learning Curve
Delayed Discovery Due to Silos
Inability to Anticipate Problems Downstream
Learning Curve Exponent
Effect of Gap on Rework
<System Understanding Gap>
Learning Curve Strength
Table of Gap vs Rework Fraction

Figure 8: Specialization Variables

3.3 Behavior Analysis

Model testing is a series of controlled experiments that test the system response to various inputs. It starts by setting the system to initialize in equilibrium in which every stock stays constant over time. In other words, all inputs and outputs are balanced. It is good practice to start under equilibrium conditions in order to isolate the true system reaction to the changing inputs.

Testing Overloaded Scenario

To replicate the current situation in PE, Equilibrium is set for one department that is overloaded with a constant work input. In recognition that something needs to be done, management triggers a reorganization at Time = 1984 and simulations are run for various degrees of specialization (DOS). When reorg is kicked off, Percent Scope of Work is divided by the DOS. Thus, DOS =1 represents one department with 100% scope of work, whereas DOS=4 represents 4 departments each with 25% of the work as shown in Figure 9.

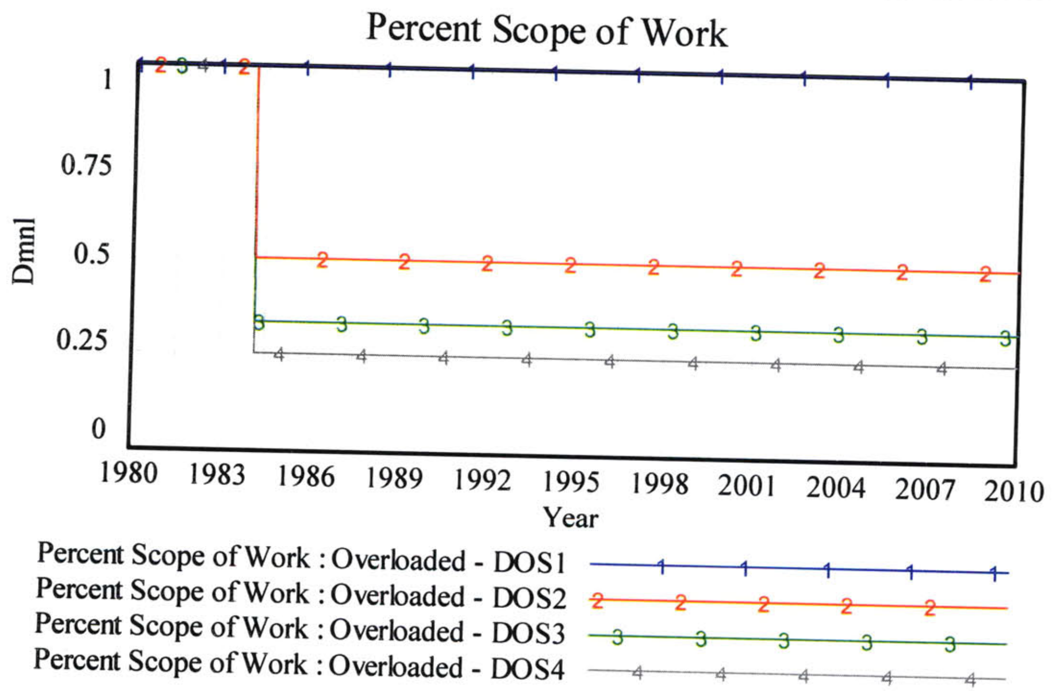


Figure 9: Specialization - Percent Scope of Work

Results from Specialization

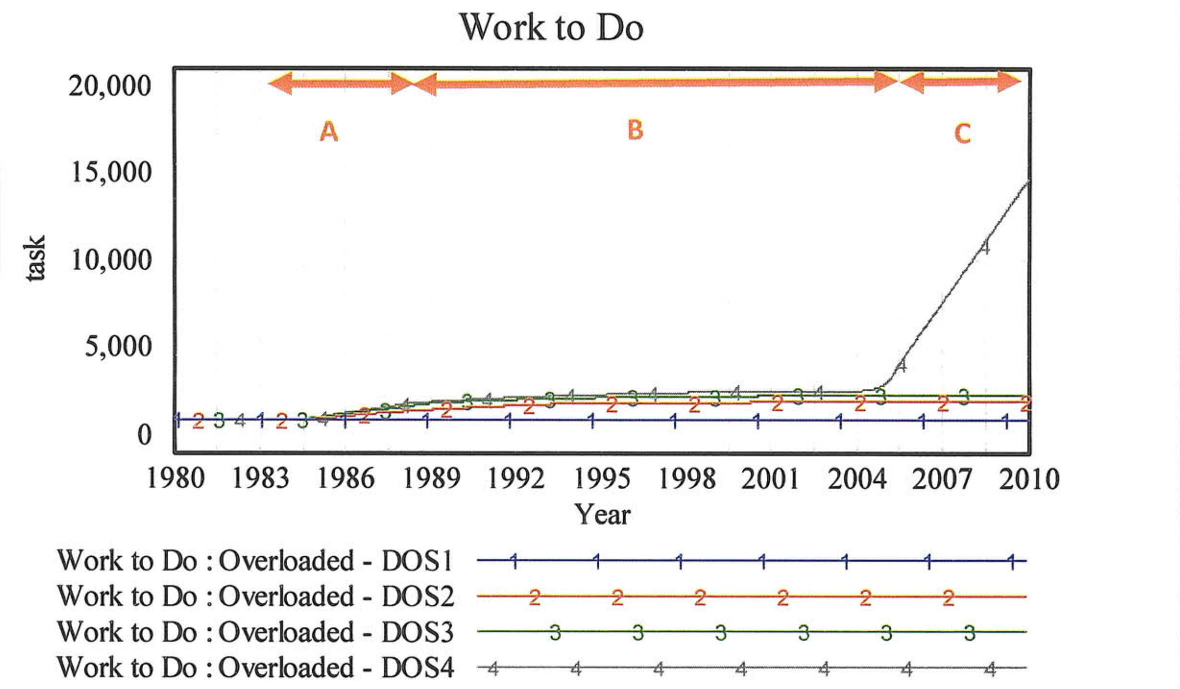


Figure 10: Specialization - Effect on Work to Do

For all cases, Work to do actually increases after specialization even though the input of new work was not changed. To facilitate the discussion, there are three phases of interest: A) Rapid growth, B) Slow growth towards a new equilibrium, C) For DOS4 only, continually increases.

Explanation for Work to Do

As Figure 10 shows, Work to Do increases after specialization, which is counterintuitive. For this to occur, inflow of work must be greater than outflow. Since Work Generation is constant for all simulations, Rework Discovery (and Rework Generation) must become greater than Work Completion Rate. Because it is a reinforcing loop, once Rework Generation is greater than Completion, work just keeps piling up, all else remaining equal.

In Periods A and B, Work increases due to an increase in Fraction Rework that is proportional to the increase in System Understanding Gap as shown in Figure 11. Because of the large initial increase of Net Productivity, the system is able to accommodate the increase of Rework.

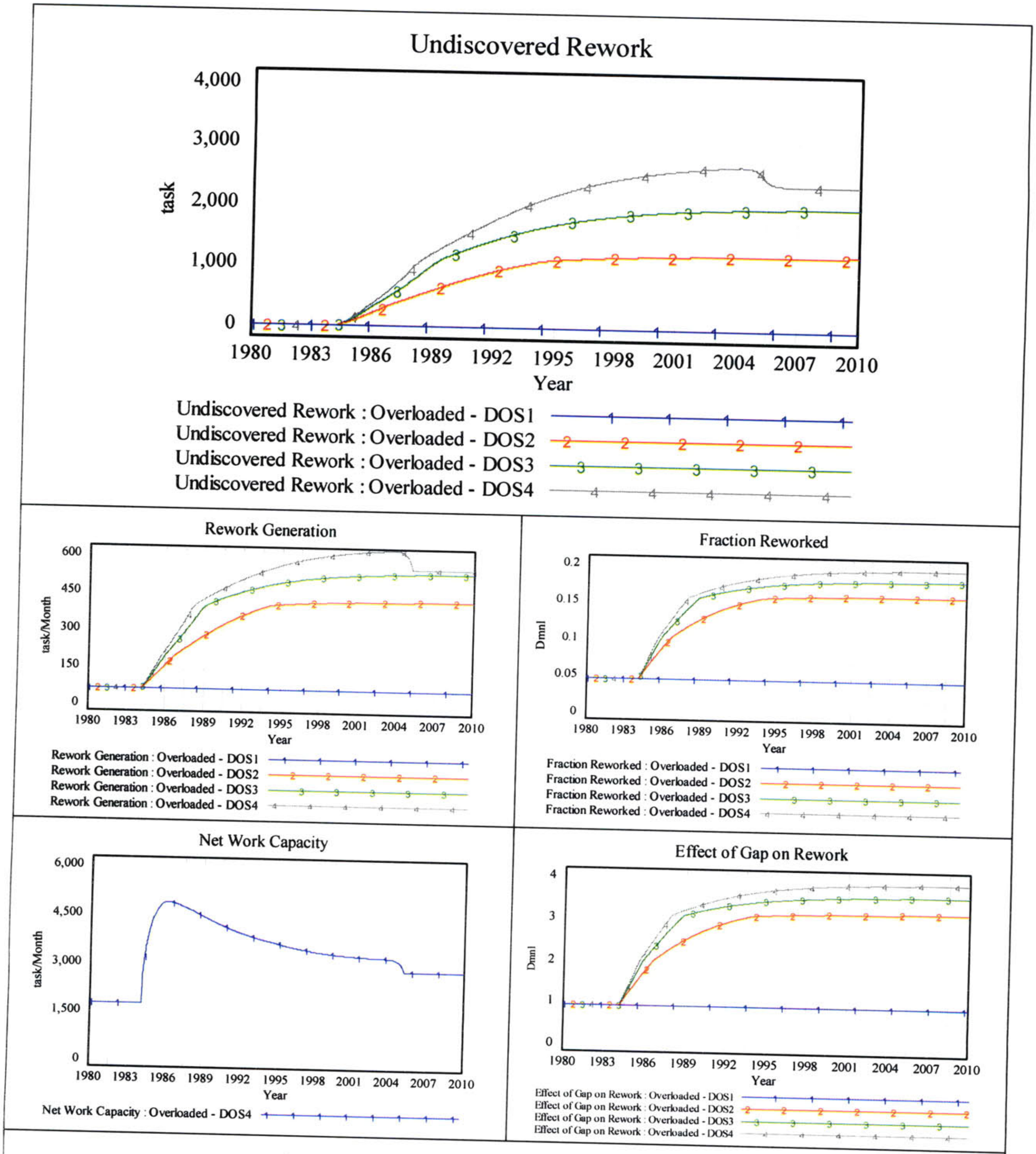
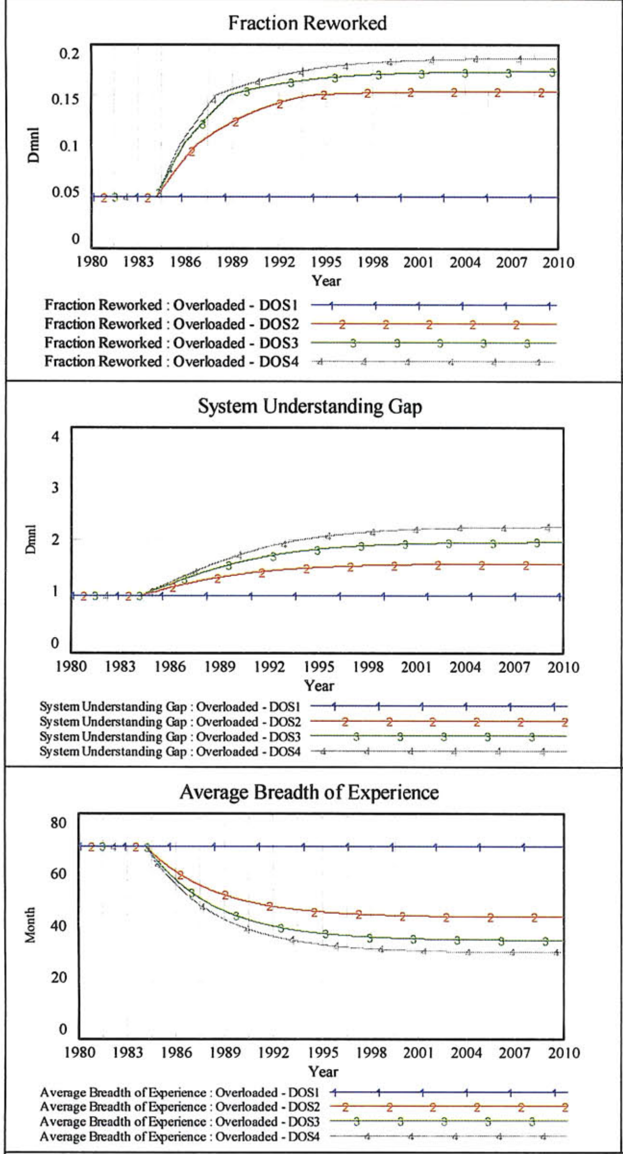


Figure 11: Specialization - Effect on Rework

Rework increases with Specialization because the Effect of Gap on Rework increases Fraction Reworked. For DOS4, the drop in Phase C is directly related to the drop in Net Work Capacity which causes Rework Generation to drop as well.



Fraction Rework

Fraction Rework increases proportionally with System Understanding Gap.

System Understanding Gap

System Understanding increases as Average Breadth of Experience diminishes.

Average Breadth of Experience

Breadth of Experience decays over a long period of time because the rate of increase is lower than the rate of decrease as shown in Figure 13.

Figure 12: Specialization - Variables Affecting Fraction Rework

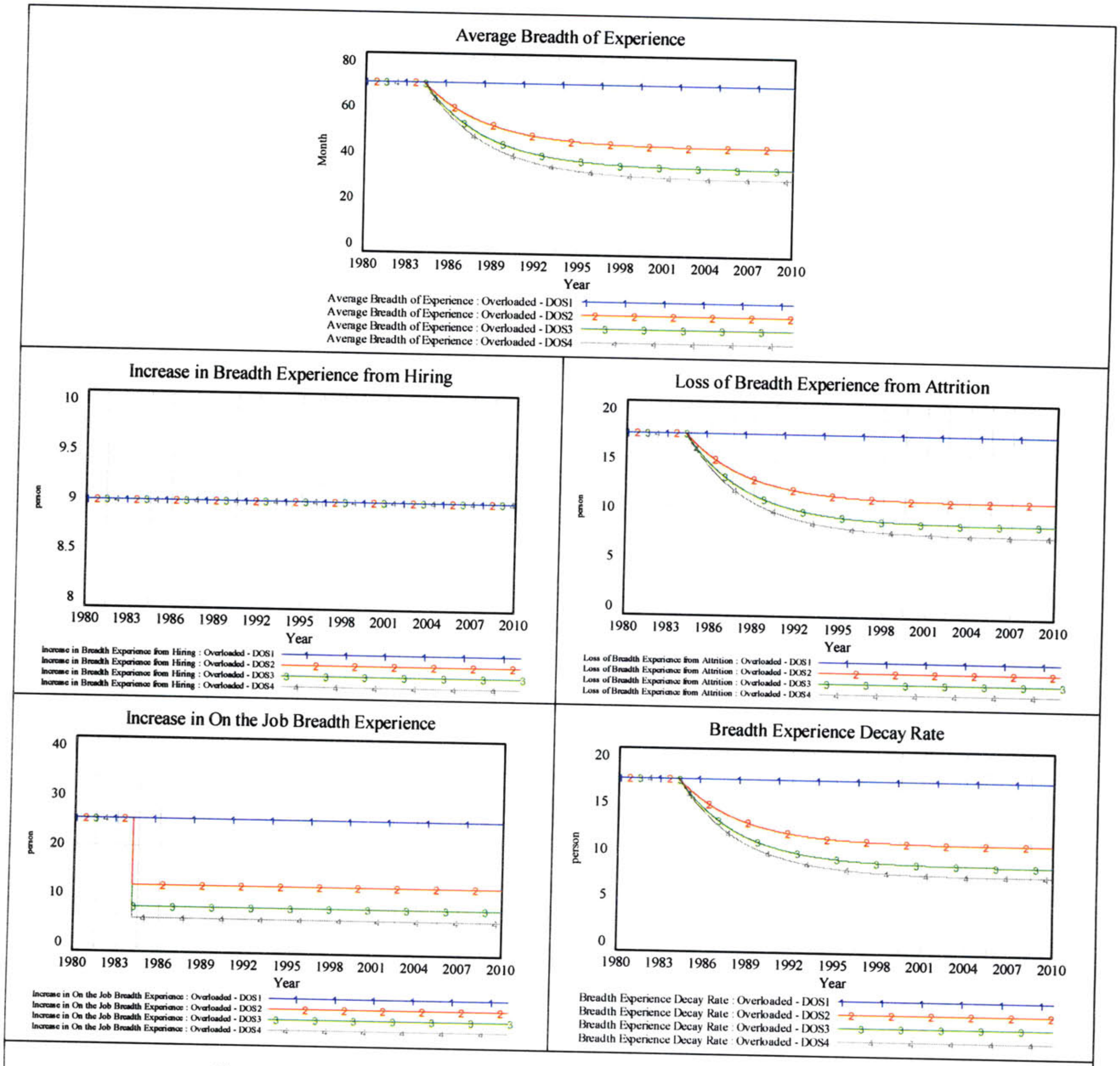


Figure 13: Specialization - Effect on Breadth of Experience

When the reorganization is triggered, the outflow (graphs on the right) becomes greater than the inflow (graphs on the left) and Breadth decreases over time until it reaches a new equilibrium.

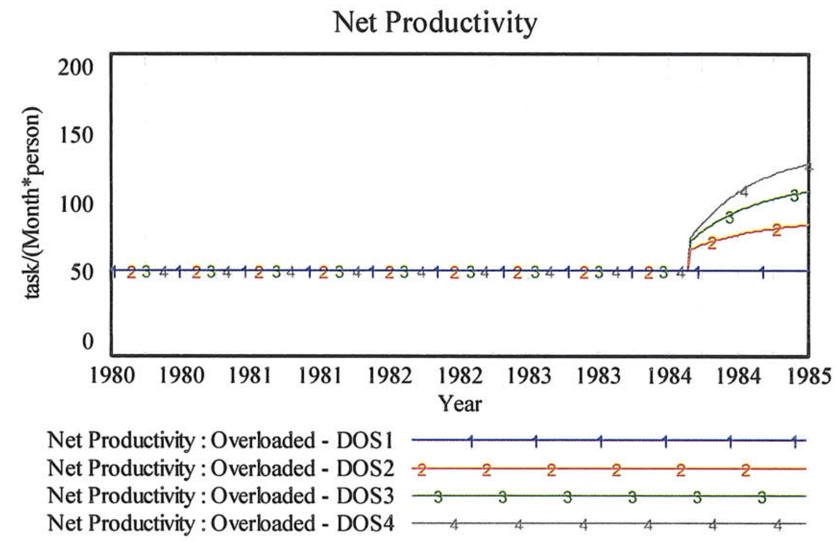


Figure 14: Specialization - Initial Net Productivity

Initial Net Productivity looks great. Often, changes or improvements are only monitored for the first few years when attention is shifted to more recent events. However, it would be misleading to only look at the first few years of Productivity because there are delayed effects 10 years later as shown in Figure 15.

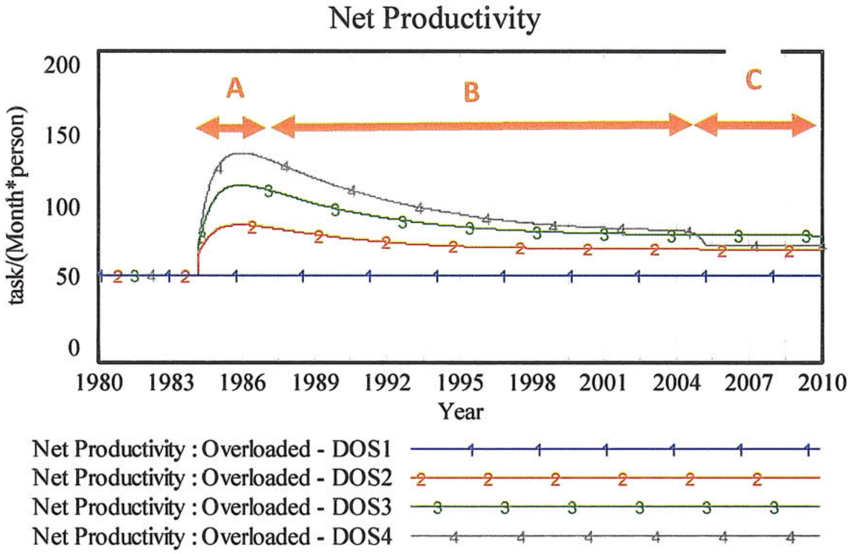


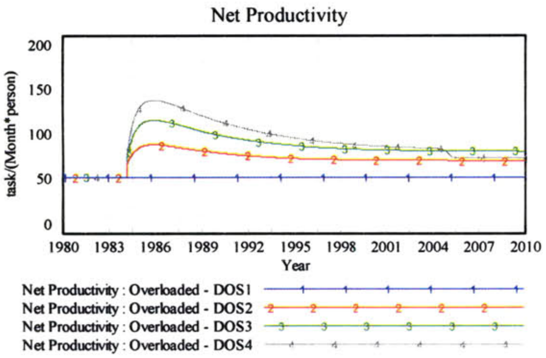
Figure 15: Specialization - Effect on Long Term Net Productivity

Net Productivity increased significantly at the point of reorganization. However, it becomes worse over time. Like Work to Do, there are three time periods of interest: A) Period of rapid increase, B) Period of slow decline towards a new equilibrium, and C) For DOS4 only, there is an additional drop.

In Period C, the sharp increase in Work to Do cannot be explained by only Rework. Here, the Multitasking Loop becomes a factor. To understand this phase, the behavior of Productivity must be understood as well.

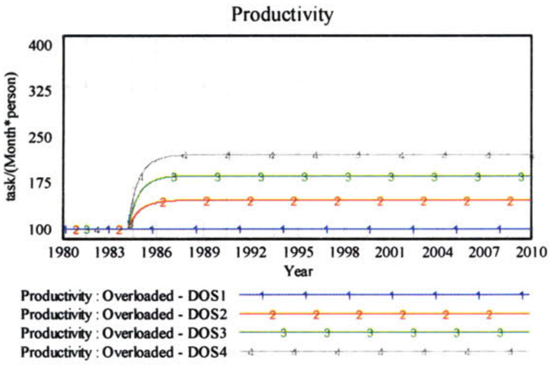
Per Figure 15, Net Productivity increases rapidly in Period A from gain in Productivity and immediate reduction of Switchloss, but slowly declines through Period B as both Coordination Cost increases and Switchloss increases due to additional workload as shown in Figure 16. The gradual pace follows the decline of Breadth of Experience and steady addition of Rework. At Time = 2005, it drops suddenly which triggers a tipping point that sends Work to Do rapidly increasing through the reinforcing Multitasking Loop.

An explanation of the tipping point can be seen in Figure 17. Because Work to Do keeps increasing from Rework and Capacity keeps dropping from Switchloss and Coordination Cost, the point where Work to Do exceeds Capacity is when inflow is greater than outflow. This triggers the rapid increase of Work to Do seen in Period C as the Multitasking Loop kicks in.



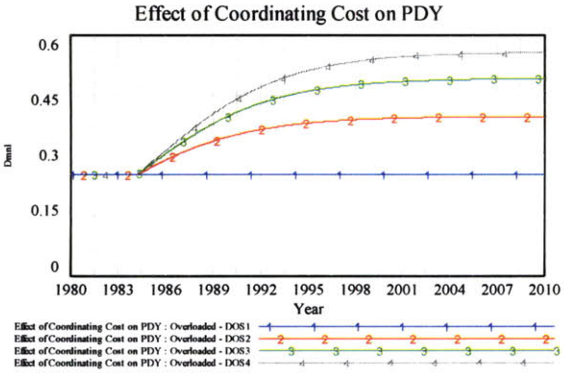
Net Productivity

Initial Productivity gains are eventually overcome by Coordinating Cost and Switchlosses.



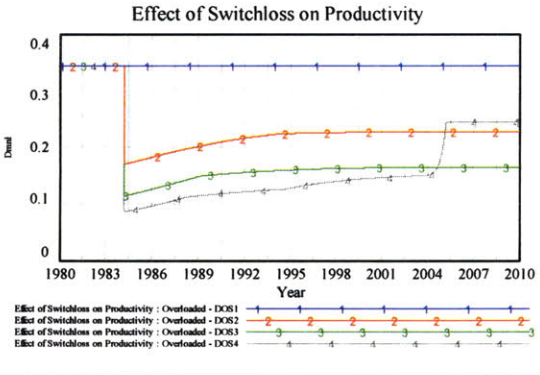
Productivity

The rapid Net growth in Period A is due to the efficiency gains from Productivity which then remains constant through Period B.



Effect of Coordinating Cost on PDY

The effect of coordination cost, though delayed, is the main factor to diminished Net Productivity.



Effect of Switchloss on PDY

Initial Switchloss reduction contributed to the Net increase during Period A. However, due to the increasing workload, Switchloss increases through Period B. For DOS4, the sharp increase in Work to Do triggers the increase in Switchloss. It does not revert back to initial levels because of the reduction in Scope of Work.

Figure 16: Specialization - Variables Affecting Net Productivity

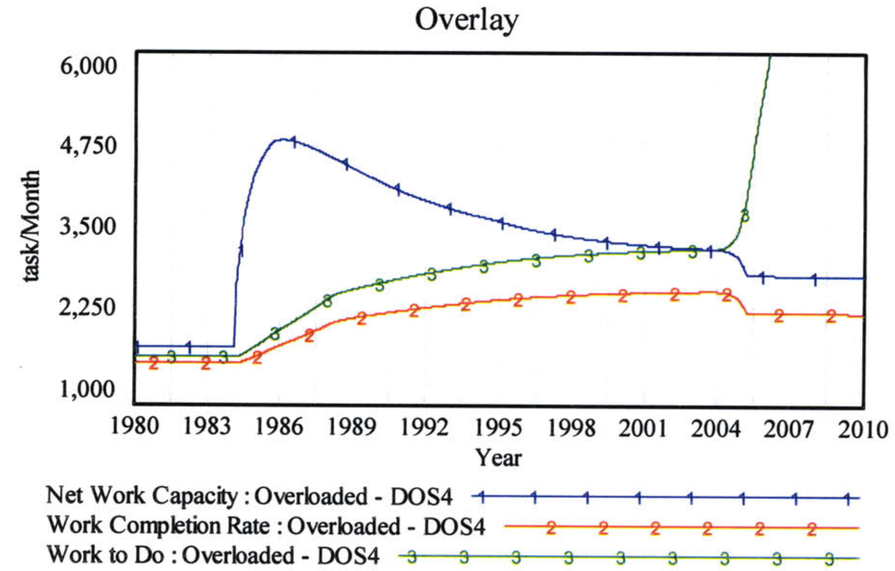


Figure 17: Specialization - Overlay of Capacity, Work to Do, and Completion Rate

Because Work to Do keeps increasing from Rework and Capacity keeps dropping from Switchloss and Coordination Cost, the point where Work to Do exceeds Capacity is when inflow is greater than outflow. This triggers the rapid increase of Work to Do seen in Period C.

In essence, initial results from Specialization look promising because there is an immediate effect of increased Net Productivity. From the structure of the model, the initial effect is because the reduction in Scope of Work and subsequent reduction of Switchloss is felt instantly. In the real world, if someone took away half the variety of tasks, there would be an instant reduction of switchloss because you are doing more of the same.

Due to the nature of the stock structures, Experience gains and decays are delayed. Though the Rate of Depth gain increases immediately, the stock does not instantaneously increase. In the model, it takes about 4 years to reach the greater Depth of Experience level and full Productivity potential. This delay is not just in the model. In real life, even if the time is cut in half, the time it takes a new hire to learn a new job can never be zero. Every new hire has to wait for their computer to be setup, get office supplies, learn how to use the phone system, get a badge, and fill out new employee paperwork, even before actually learning about the specific job.

In contrast to Rate of Depth Increase, Rate of Breadth Increase goes down due to Specialization. As a result, Average Breadth of Experience declines over a 20 year period before reaching a new equilibrium. As it declines, the amount of Rework increases and Net Productivity decreases due to coordination cost, both of which leads to increased Loading per Person that eventually triggers the Multitasking Loop.

(This page intentionally left blank.)

4 Reinventing the Wheel Model

The initial title of this thesis was "Is specialization the solution to a burgeoning product portfolio?" because I came into the project with the mindset that the overloading of the department was due to an exogenous inflow of projects that the department, Product Engineering, had no control over. This is the prevailing wisdom and potential attribution error. After talking to people and thinking about the problem further, I came to the realization that some of the overload is in fact self-inflicted. This model is an effort to explore alternative explanations as to why we are overloaded in the first place. Granted, the burgeoning product portfolio is a major factor in setting initial conditions, but internal behaviors have exacerbated the problem.

The inspiration for the Reinventing the Wheel Model is a design problem that was identified 10 years ago shortly after a new product launch. Though the item originated as a rework item, the story here is about the number of times that it has been handed off from one person to the next and how much additional work has been generated due to reinventing the wheel. It just keeps churning without resolution because it has passed so many hands. Each person adds some information to the problem, but that means it will take incrementally more time for the next person to get up to speed which means that their contribution will be even smaller assuming the same duration of ownership. Everyone in the department has "owned" this project at one time or another.

4.1 Causal Loop

Figure 18 describes Reinventing the Wheel through a causal loop. Reading it clockwise, as Time to Complete increases, # of Handoffs from person to person increases, Knowledge Loss increases, Additional Work Generated increases due to reinventing what others had already done, which increases Cumulative Work to Do and ultimately increases Time to Complete. Because Time to Complete increases with every pass of the cycle, it is a reinforcing loop.

There are two factors to the reinventing causal loop that we will explore: those that effect time to complete and the different types of handoffs.

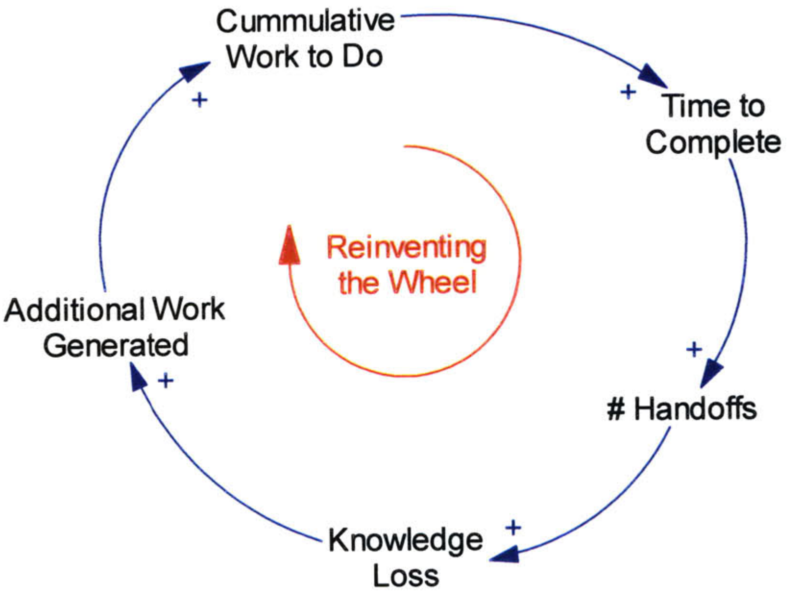


Figure 18: Reinventing the Wheel Causal Loop

Handoff Types

1. Turnover – People leaving the department or the company.
2. Reshuffling - Depending on the hot projects at the time, work is constantly getting reprioritized and shuffled from person to person every six months. This was not modeled explicitly because the system behavior is going to be very similar to the turnover rework.

Handoff type determines the level of knowledge loss and the amount of additional work generated. For example, turnover is the worst case scenario because it applies to people leaving the department or company which means a higher portion of knowledge is lost leading to higher reinventing. Reshuffling of work from person to person still in the department means that the person is still around to share their knowledge with whoever is taking over, so less knowledge is lost.

Why does it take so long to complete everything?

1. Multitasking – Everyone is overloaded and trying to juggle many projects all at once. This reduces productivity which means it takes longer to complete the same tasks.
2. Prioritization – Low priority work falls to the bottom of the list and never gets attention. In any overload situation, people typically go into triage mode and only work on the most critical.

For simplicity, I have chosen to only model multitasking and turnover to illustrate that any delay in work completion will invoke the reinventing the wheel loop. Adding the other contributors would only amplify the situation. Whether the handoff is due to turnover or reshuffling, the effect is the same: more work generated. Whether it is due to multitasking or prioritization, the effect is the same: it takes longer to finish a project.

4.2 Developing the Structure

4.2.1 Baseline Work Structure

There are two loops within the resource allocation and work structure. Starting with Work to Do, if that increases, Weight of Work to Do increases, Fraction Capacity to Partially Complete increases, Partial Completion Rate increases, and Work to Do decreases. This is a balancing loop. Work Partially Complete has the same structure. Work Capacity is split according to the Fraction Capacities. If one goes up, the other goes down.

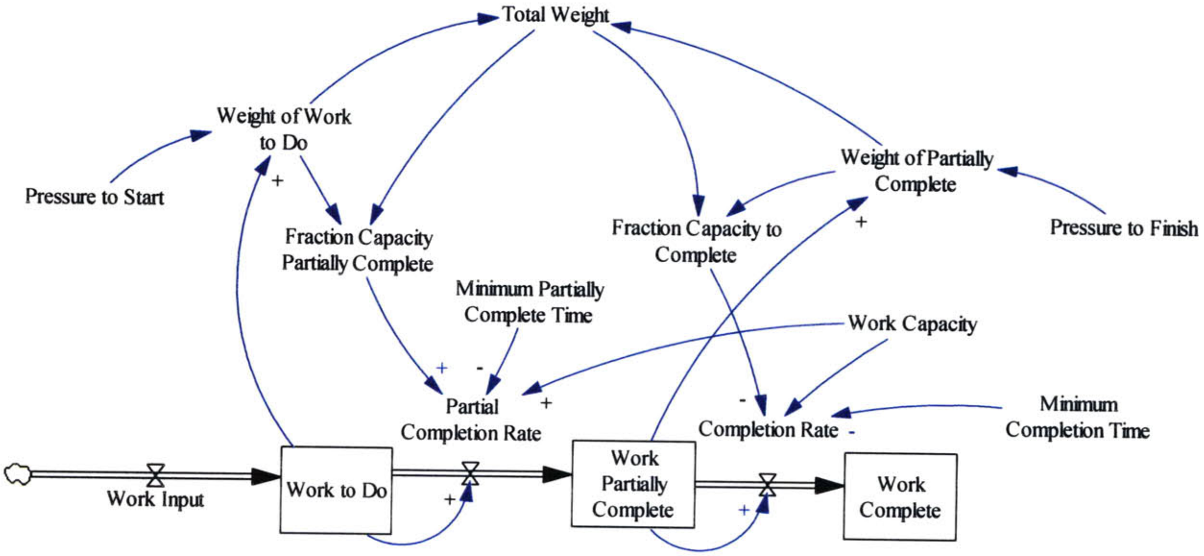


Figure 19: Baseline Work Structure

Key Variables
Key variables are qualitatively described in detail within the thesis. For all others, see Appendix B.

Work to Do, Work Partially Complete and Work Done are the basic stocks of interest. Work flows into the system through *Work Input*. Work to Do passes into Work Partially Complete at the *Partial Completion Rate*. Work Partially Complete passes into Work Done at the *Completion Rate*, which comes from Work Capacity. Work to Do represents early stages of work where tasks are just recently assigned, but minimal effort has been put into them. Tasks in Work Partially Complete have had some attention and effort invested into them, but not yet complete.

Pressure to Start is used to add managerial pressure on the system to increase resources assigned to tasks in Work to Do.

Pressure to Finish is used to add managerial pressure on the system to increase resources assigned to complete tasks in Work Partially Complete. Under equilibrium conditions, it is equal to Pressure to Start.

Weight of Work to Do is the product of the Work to Do stock and the Pressure to Start. As Work to Do increases, it demands more attention from resources.

Weight of Work Partially Complete is the product of Work Partially Complete and the Pressure to Finish. As Work Partially Complete increases, it demands more attention from resources.

Total Weight is the sum of Weight of Work Partially Complete and Weight of Work to Do.

Fraction Capacity to Partially Complete is the product of Work to Do and Pressure to Start divided by Total Weight. It determines how much Work Capacity is allocated to Partially Complete tasks.

Fraction Capacity to Complete is the product of Work Partially Complete and Pressure to Finish divided by Total Weight. It determines how much of Work Capacity is allocated to complete tasks.

Work Capacity is constant in the basic structure. It is split between the Completion Rates depending on Fraction Capacity to Partially Complete and Fraction Capacity to Complete. In baseline conditions, the system starts with capacity equally split between the two rates.

Minimum Time to Complete is the least amount of time it can take to complete the task.

Minimum Time to Complete Partially is the least amount of time it can take to complete the task.

Normalized Variables Input and Work to Do are normalized so that changes in behavior can be judged relative to starting conditions.

Behavior Analysis

For the system to be in equilibrium, Work to Do must equal Work Partially done.

Test Scenario: Pulse Input

Baseline conditions are set at 75% of work capacity so that everyone is 75% loaded.

A pulse input is a good way to test the system. It tracks a one time input of tasks through to completion. If it cannot recover from a pulse of work, then there is no way for it to recover from a step or ramp change.

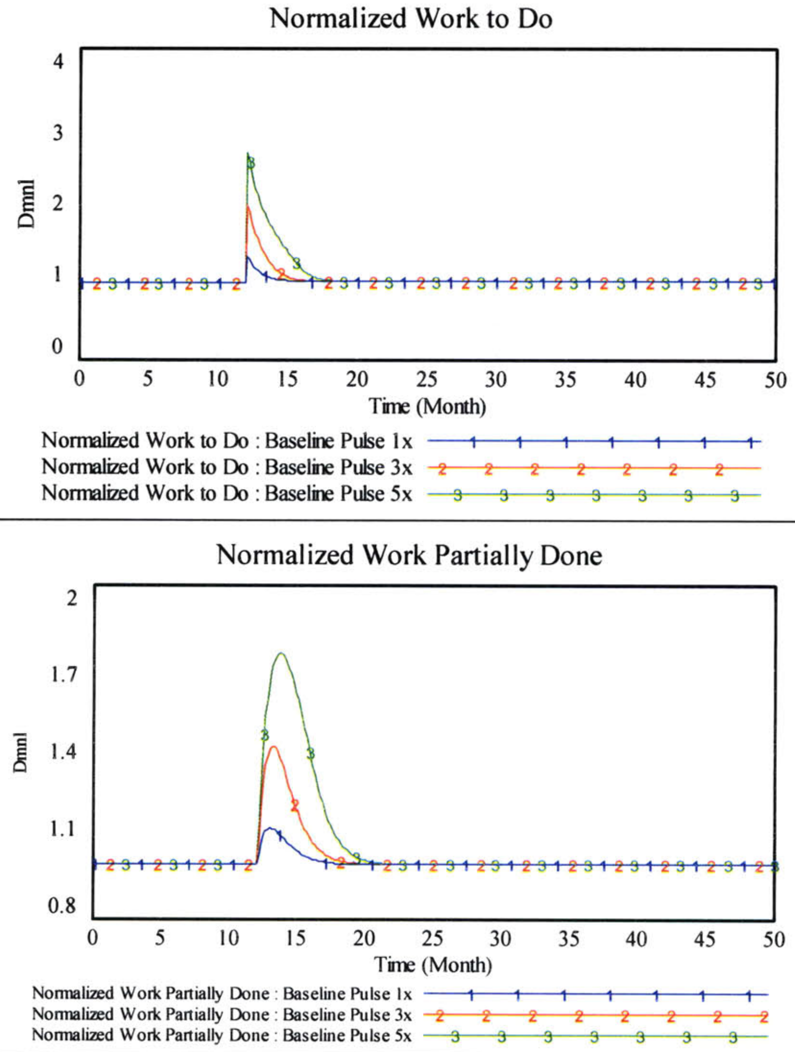


Figure 20 Baseline – Effect of Varying Pulse Input

Pulse Inputs of increasing height were triggered at Time=12. The Baseline structure recovers after 5 to 10 months. Subsequent tests will compare to this Baseline response to varying pulse inputs.

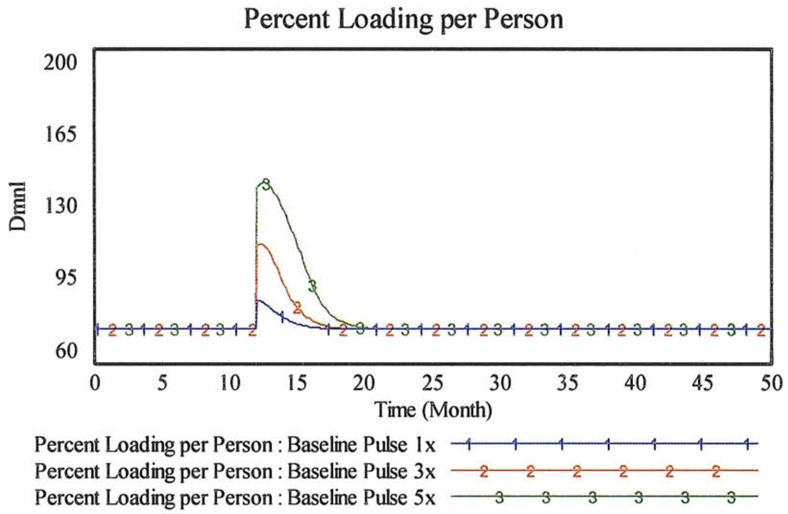


Figure 21: Baseline – Effect of Pulse Input on Percent Loading per Person

In the Baseline scenario, Percent Loading per Person has no impact to the system because there is no feedback mechanism.

Because the initial equilibrium workload is lower than Work Capacity, the system has excess capacity to respond and recover from unanticipated work even if the Percent Loading is high (Figure 21). The system eventually recovers as long as Work Input does not exceed Work Capacity.

Test Scenario: Sensitivity to Initial Conditions

This set of tests explores sensitivity to starting conditions.

When the system starts with Input already at 100% Loading, there is no extra capacity for anything else. Everyone is already working full out on existing incoming tasks. That means either the pulse of work never gets done, or more realistically, high priority items get done, but the low priority items linger forever.

The implication here is that if an organization is already fully loaded, adding anything else will mean that something is not getting done and there is always a deficit. The system cannot recover from the slightest increase in work input. Figure 22 illustrates how sensitive the system is to initial Percent Loading.

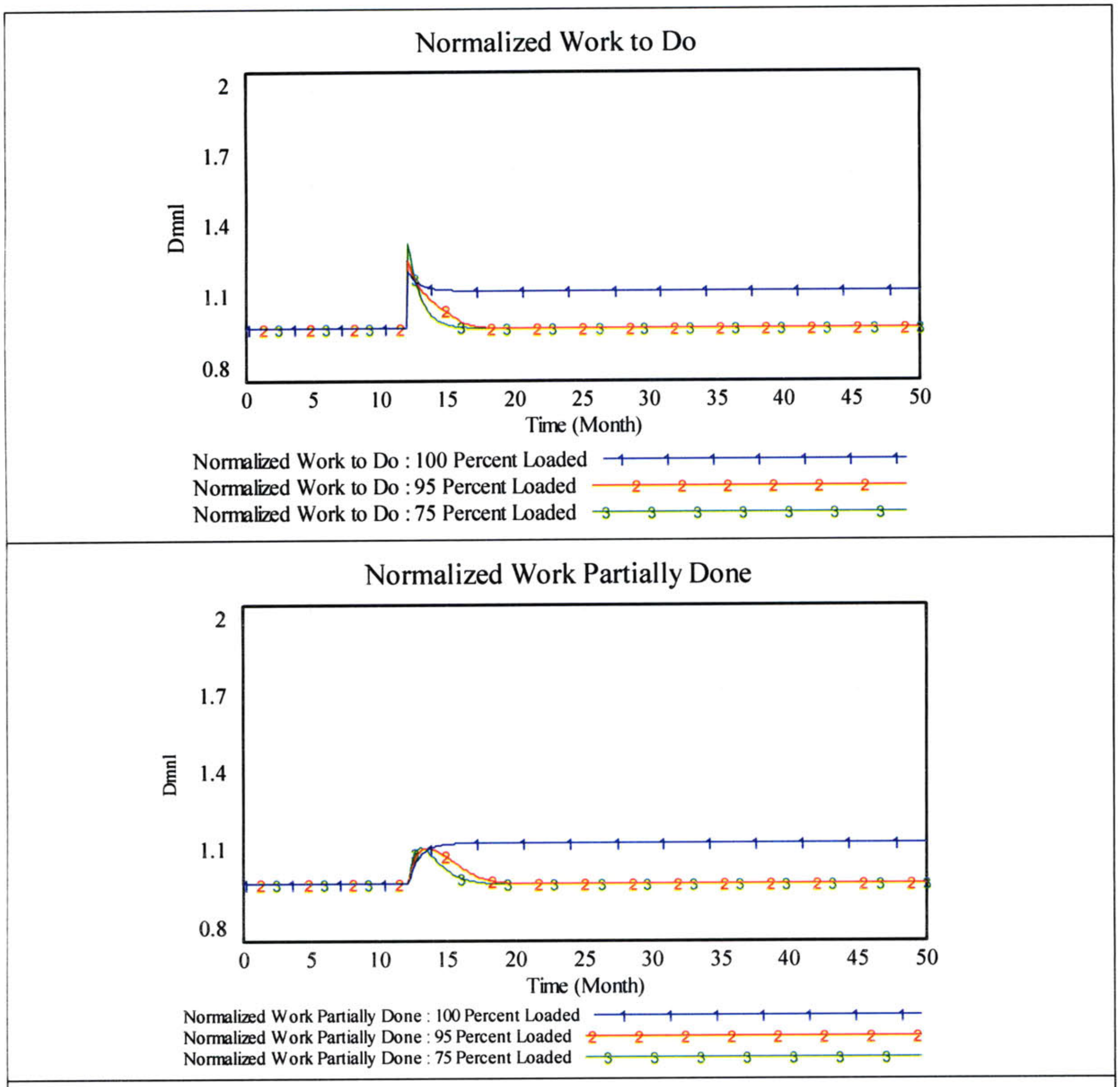


Figure 22 Baseline – Effect of Input Loading

These three scenarios start at different equilibrium levels. It shows that if the system starts out 100% loaded [1], any input, even a one time pulse causes a permanent step change in workload that is never caught up. Even at 95% loaded [2], the system eventually recovers even if it takes longer than the 75% run [3].

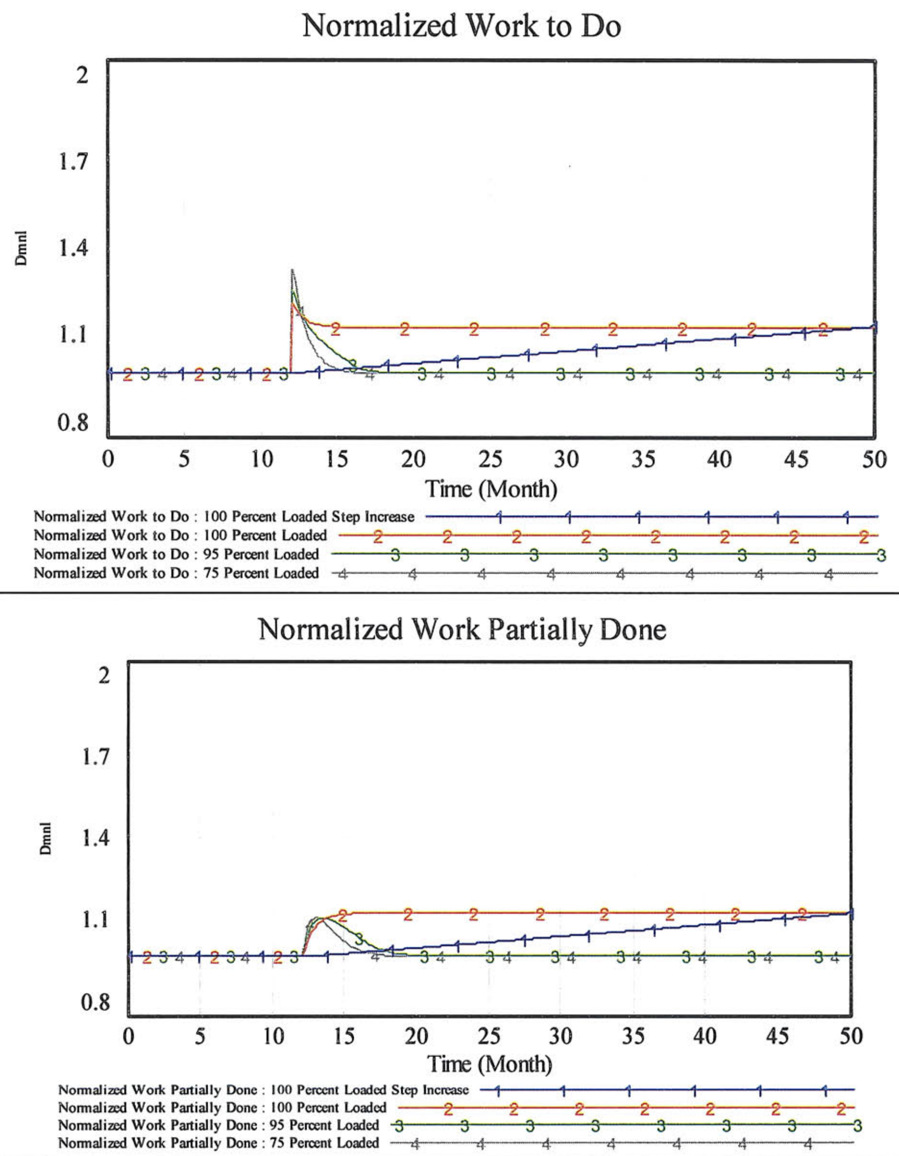


Figure 23 Baseline – Effect of Step Input

With initial loading at 100%, a step increase of only 1 task above capacity leads to a continuous increase [1].

The step increase scenario [1] in Figure 23 could explain the increasing deficit of work over the past decade with the increased introduction of new products without increasing headcount.

4.2.2 Overtime Loop

Basis

Up to now, initial conditions were set at underloaded levels to provide the system with extra capacity to respond to increased input. Another way of providing excess capacity is to add Overtime Pressure. In the real world, there is some elasticity in work capacity. Given an increased workload, people will tend to put in overtime and "do whatever it takes" to finish the job.

Structure

The Overtime Loop runs counterclockwise. Starting with Work per Person, if it increases, Percent Loading per Person increases, Overtime Pressure increases, Additional Capacity increases, Work Capacity increases, Work Completion Rate (similarly, Partial Completion Rate) increases, Work Partially Complete (Work to Do) decreases, Total Work decreases and Work per Person decreases. Because Work per Person decreases with every pass, all else equal, it is a balancing loop. A balancing loop will adjust itself toward equilibrium.

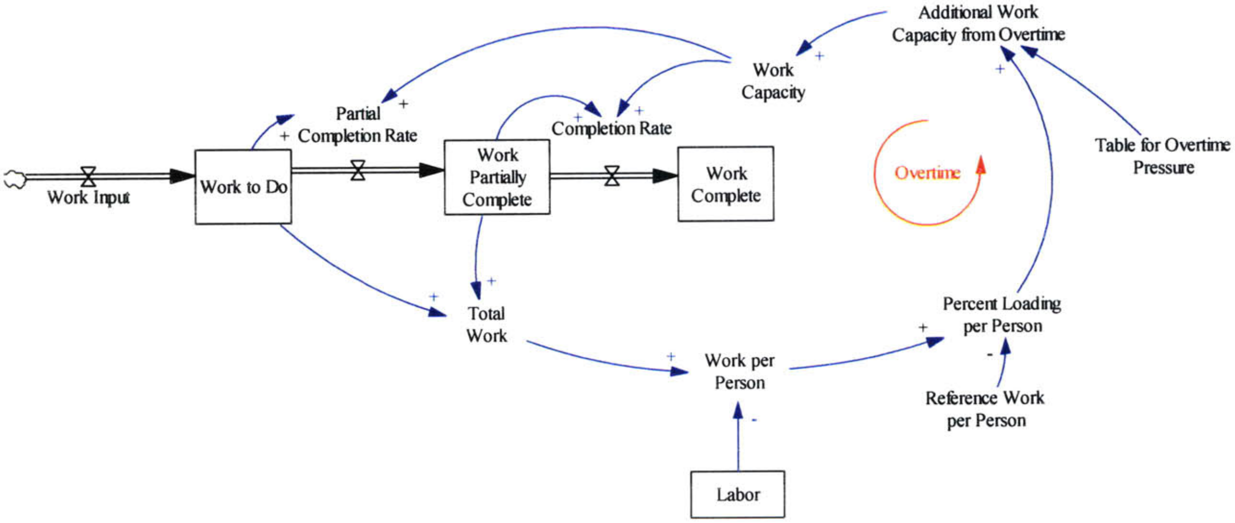


Figure 24: Overtime Structure

Key Variables

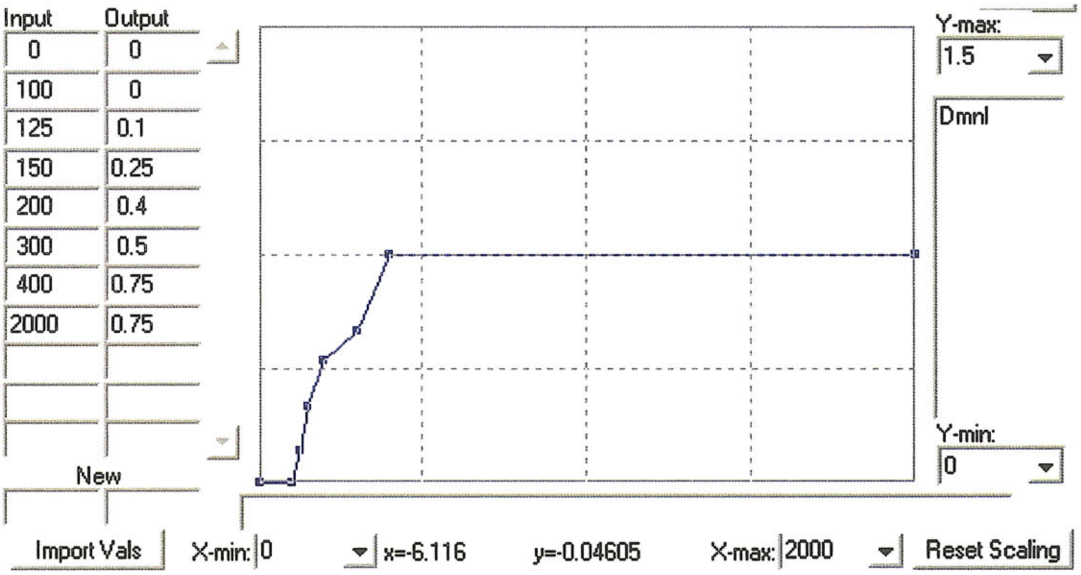
Total Work is the sum of Work to Do and Work Partially Complete.

Work per Person is a function of Total Work divided by Labor.

Labor is the stock of workforce available.

Percent Loading per Person is determined by Work per Person divided by a Reference Work per Person. The *Reference Work per Person* represents someone who is 100% loaded working 40 hours per week.

Additional Work Capacity from Overtime is determined by the *Table for Overtime Pressure* shown in Figure 25. As workload increases, the typical response is to put in overtime to complete the work. However, there is a limit to how much overtime is feasible. It is possible to work 80 hours per week, but it is not sustainable. On the other hand, 60 hours per week has become normal in the department. In the model, the maximum overtime is 70 hours per week.



Input	Output
0	0
100	0
125	0.1
150	0.25
200	0.4
300	0.5
400	0.75
2000	0.75

Y-max: 1.5

Dmnl

Y-min: 0

X-min: 0

x=-6.116

y=-0.04605

X-max: 2000

Figure 25: Table for Overtime Pressure

Behavior Analysis

Test Scenario: Baseline Conditions with Pulse Input

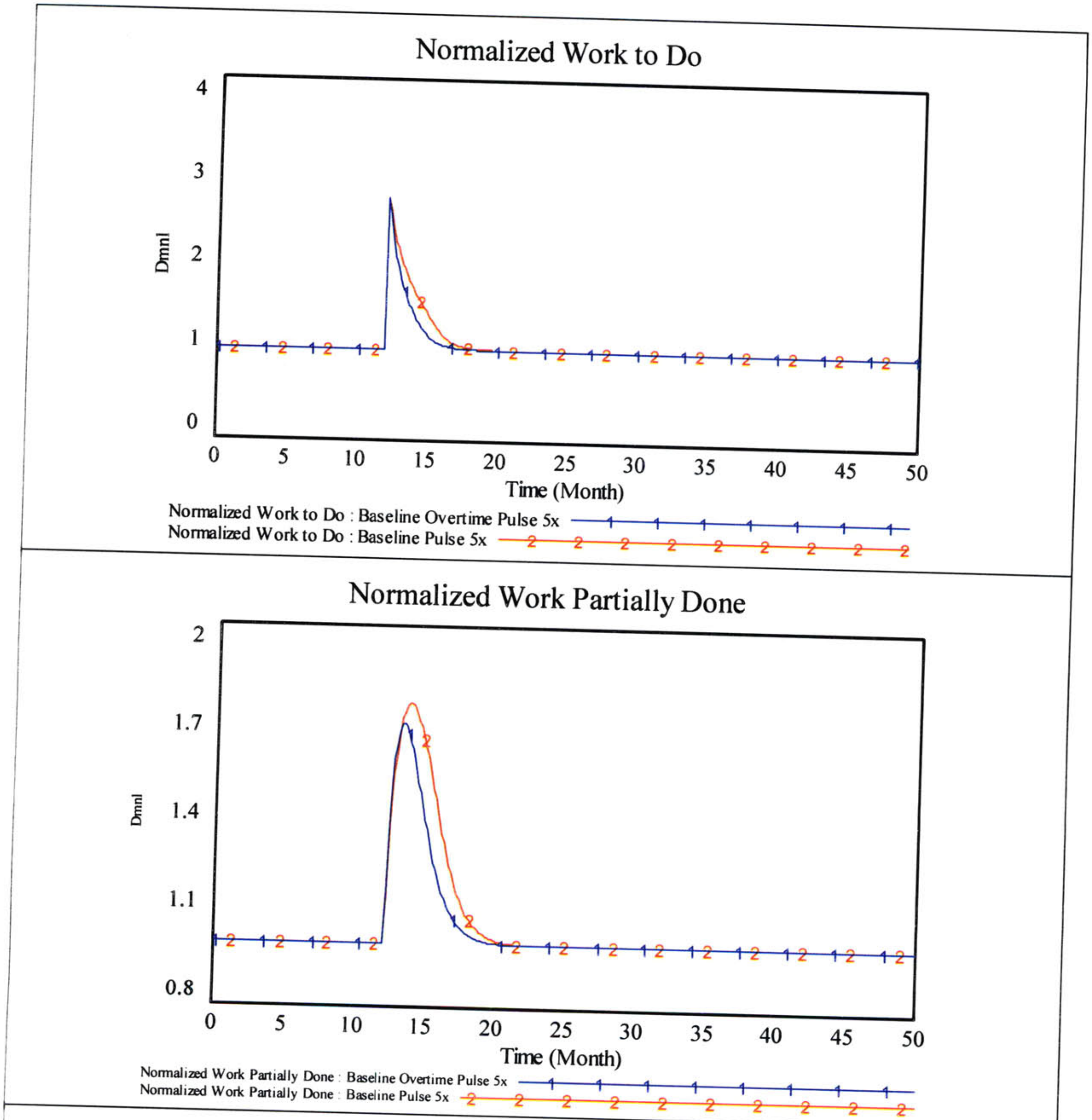


Figure 26 Overtime - Effect on Pulse Input

With Overtime activated, the pulse of work is completed faster.

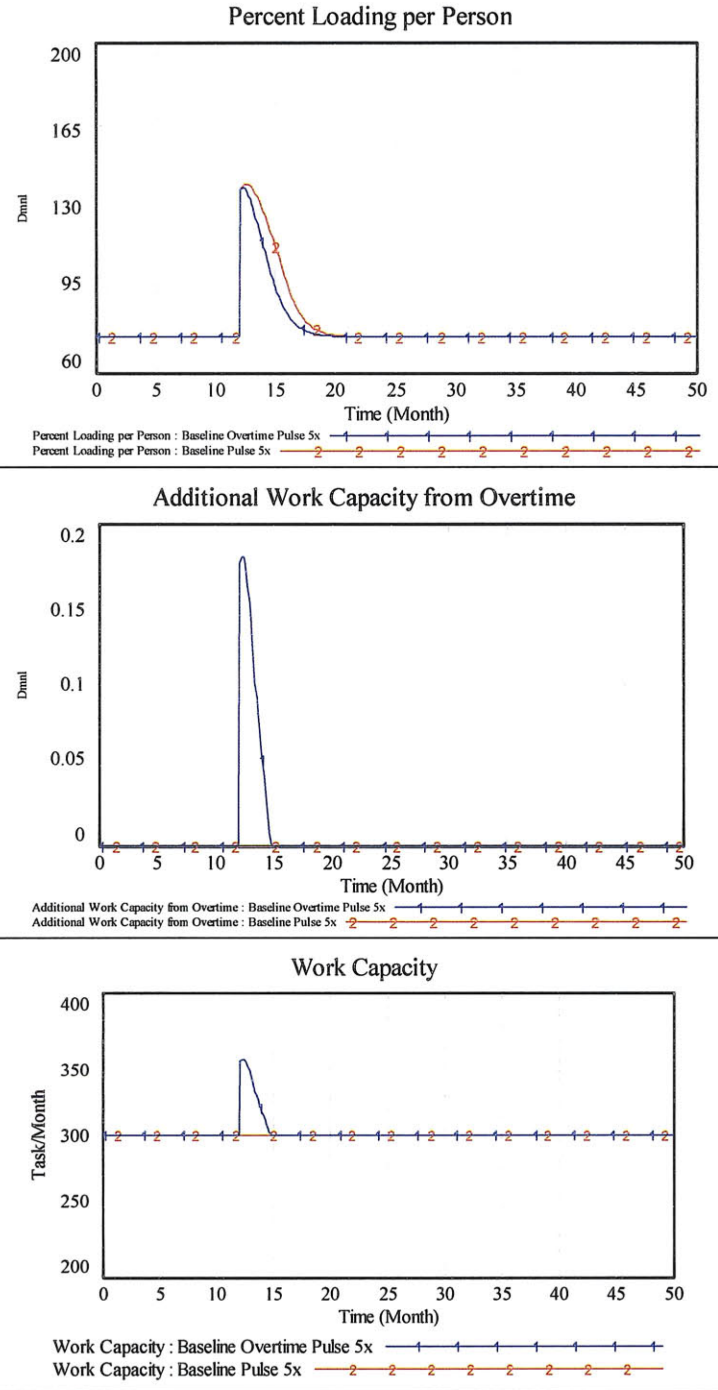


Figure 27: Overtime Causal Strip

Following the Overtime Loop, the pulse of work spiked Percent Loading per Person which enacted Additional Work Capacity from Overtime which increased Work Capacity [1]. In the Baseline run, Percent Loading had no effect or feedback to the system [2].

4.2.3 Multitasking Loop

"To do two things at once is to do neither."
- Publilius Syrius, Roman Philosopher

Basis

One complaint about PE is that it takes so long for projects to be complete. One explanation for this is switchloss. It is agreed that Product Engineering is overloaded. Under these conditions, each person has multiple projects each at varying stages and is constantly juggling tasks on a day to day basis. Generally speaking, multitasking allows the incremental progression of several projects, but the completion of none. Completing none is the case in the model because of the aggregate nature of modeling, but in reality, the high priority work gets completed and the low priority work tends to churn forever.

Often, incoming workload is out of Product Engineering's control. There is little leverage for a middle manager in the department to stave off the wave of new projects that are determined by many players "higher up" in the hierarchy. No one wants to be the one who does not deliver the expected business growth. Therefore, PE is stuck with the workload that is given and must show progress on every front for fear of being labeled the bottleneck of the business. The result is that all projects big and small are assigned to someone and, unfortunately, this means that corners are cut and tasks left incomplete before engineers are moved on to the next high priority project.

Research has shown that on average, workers are interrupted every 11 minutes and lose 2 hours a day due to switchlosses. (Crenshaw 2008). Switchloss is the time it takes to switch mental gears between tasks. Switchloss also includes the time it takes for someone to review what was done before resuming work. It is reading a half written email over and over again in between interruptions before getting the time to finish a reply. The more complicated the task, the greater the loss. Because there are not enough long stretches of time to finish a task in one sitting (before heading to a meeting or getting a phone call), whether it is writing an email or reviewing drawings or analyzing data, the engineer must review what was done 5 hours (or days) earlier before resuming the project. In the case for Product Engineers, the loss is not just switching between tasks but between projects as well. When a typical engineer is juggling 1 breakthrough, 2 derivatives, and 5 maintenance projects, all at different phases, there is a great deal of daily switching. PE's are lauded as great multitaskers, but the track record of not closing projects may tell a different story.

Structure

The multitasking loop goes counterclockwise. Starting at Work to Do, if Work increases, Work per Person increases, Percent Loading per Person increases, the Effect of Switchloss increases, Productivity decreases, Work Capacity Decreases, Work Completion Rate decreases, and Work to Do increases. Because Work to Do increases, it is a reinforcing loop which means, if all else remains equal, Work to Do will continue to increase indefinitely.

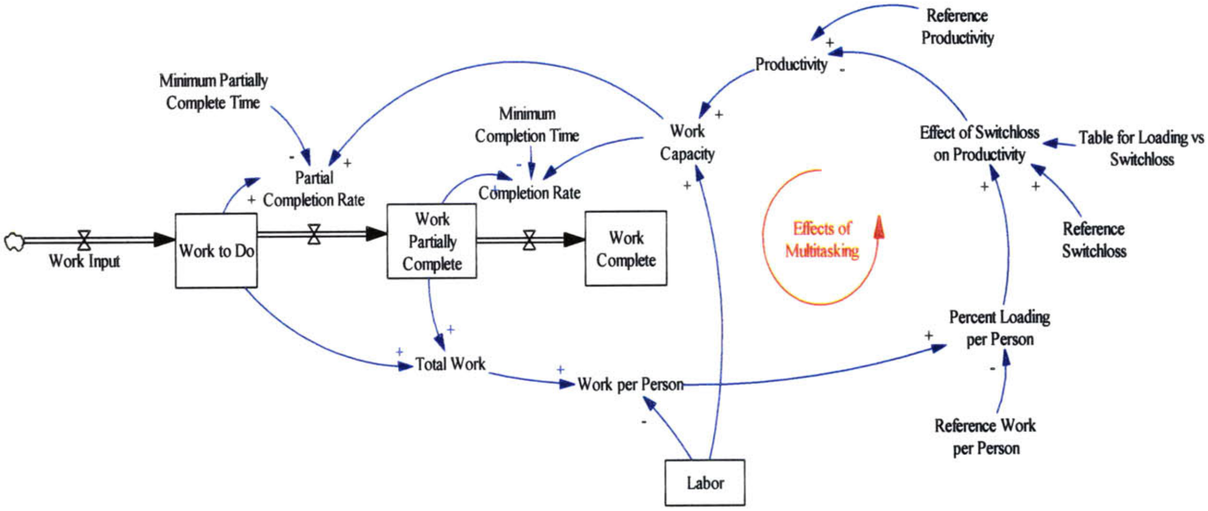


Figure 28: Multitasking Structure

Key Variables

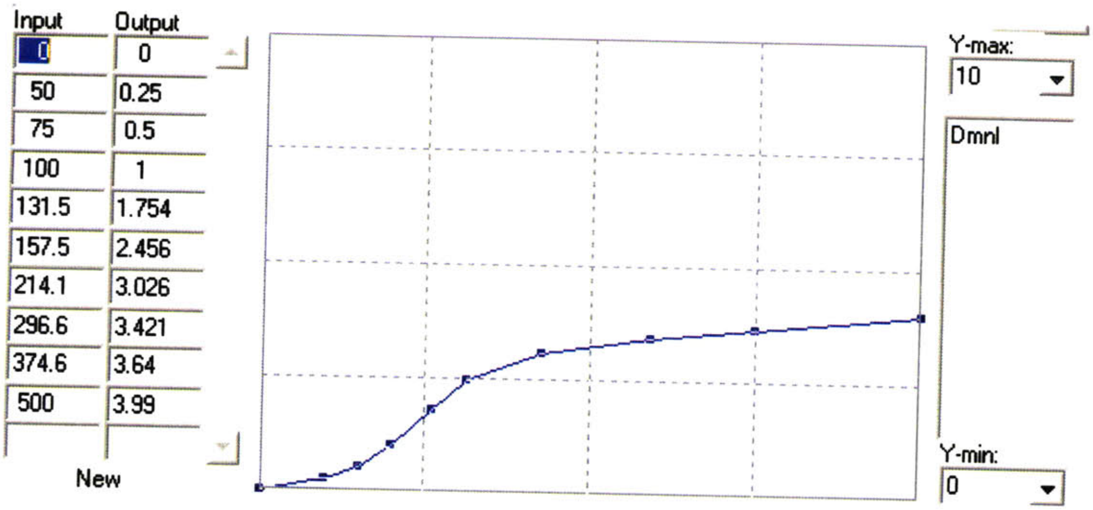
Only key variables are described in detail in the thesis. For all others, see Appendix B.

Percent Loading per Person was introduced earlier to trigger Overtime. When the Multitasking Loop is activated, Loading per Person is also used to determine Switchloss. The more people have on their plate, the more often they have to go back and forth between projects. As Loading increases, so does Switchloss per Figure 29 .

Effect of Switchloss on Productivity

As Percent Loading per Person increases, Effect of Switchloss on Productivity increases due to the constant shifting from one project and task to another. This effect takes the shape of an s-curve where it gradually increases until there is a sharp increase and then plateaus toward 100% productivity loss per Figure 29.

Reference Switchloss is set at 0.25 because studies have shown that the average knowledge worker loses 25% of their day to switchlosses (Crenshaw 2008).



Input	Output
0	0
50	0.25
75	0.5
100	1
131.5	1.754
157.5	2.456
214.1	3.026
296.6	3.421
374.6	3.64
500	3.99

New

Y-max: 10

Dmnl

Y-min: 0

Figure 29: Table for Loading vs. Switchloss

Productivity is the number of tasks per unit of effort (person*month).

Work Capacity is the product of Labor and Productivity. Up to now, it was held constant.

Behavior Analysis

Activating Switchloss

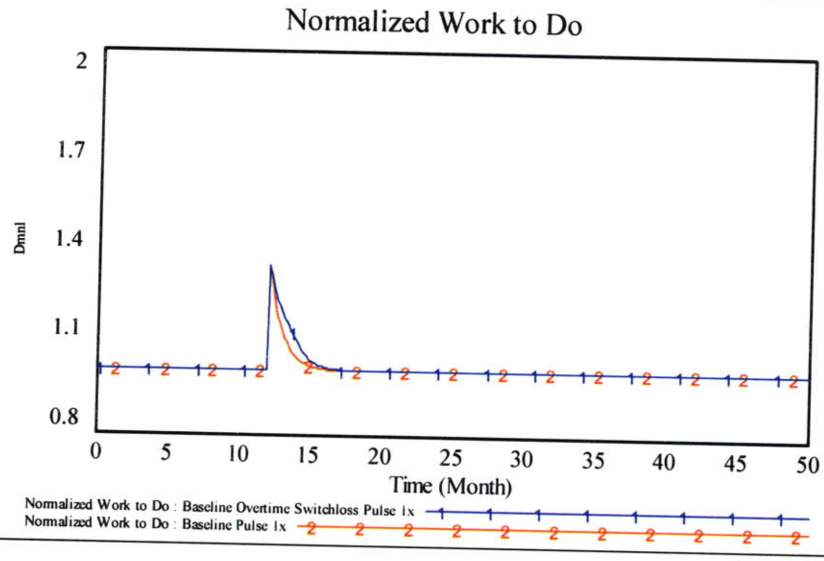


Figure 30 Multitasking - Effect on Pulse Input

Comparing the Baseline only [2] and Switchloss [1] simulations, activating switchloss increased the time it takes to complete the pulse of work.

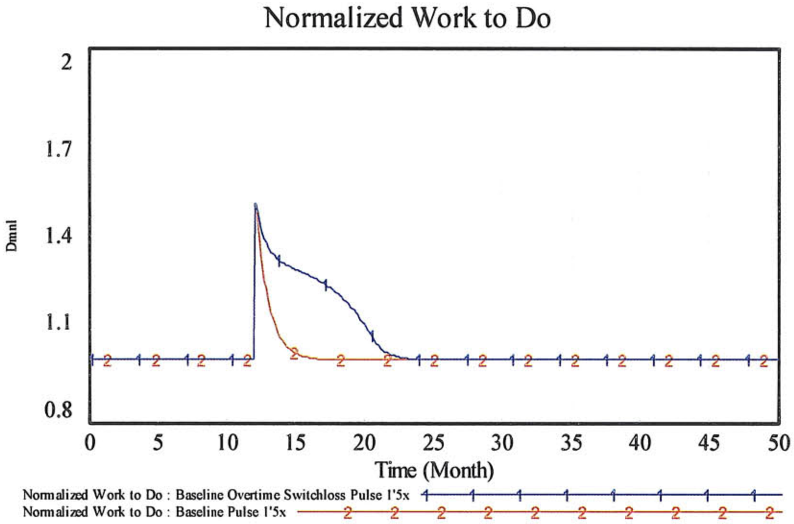
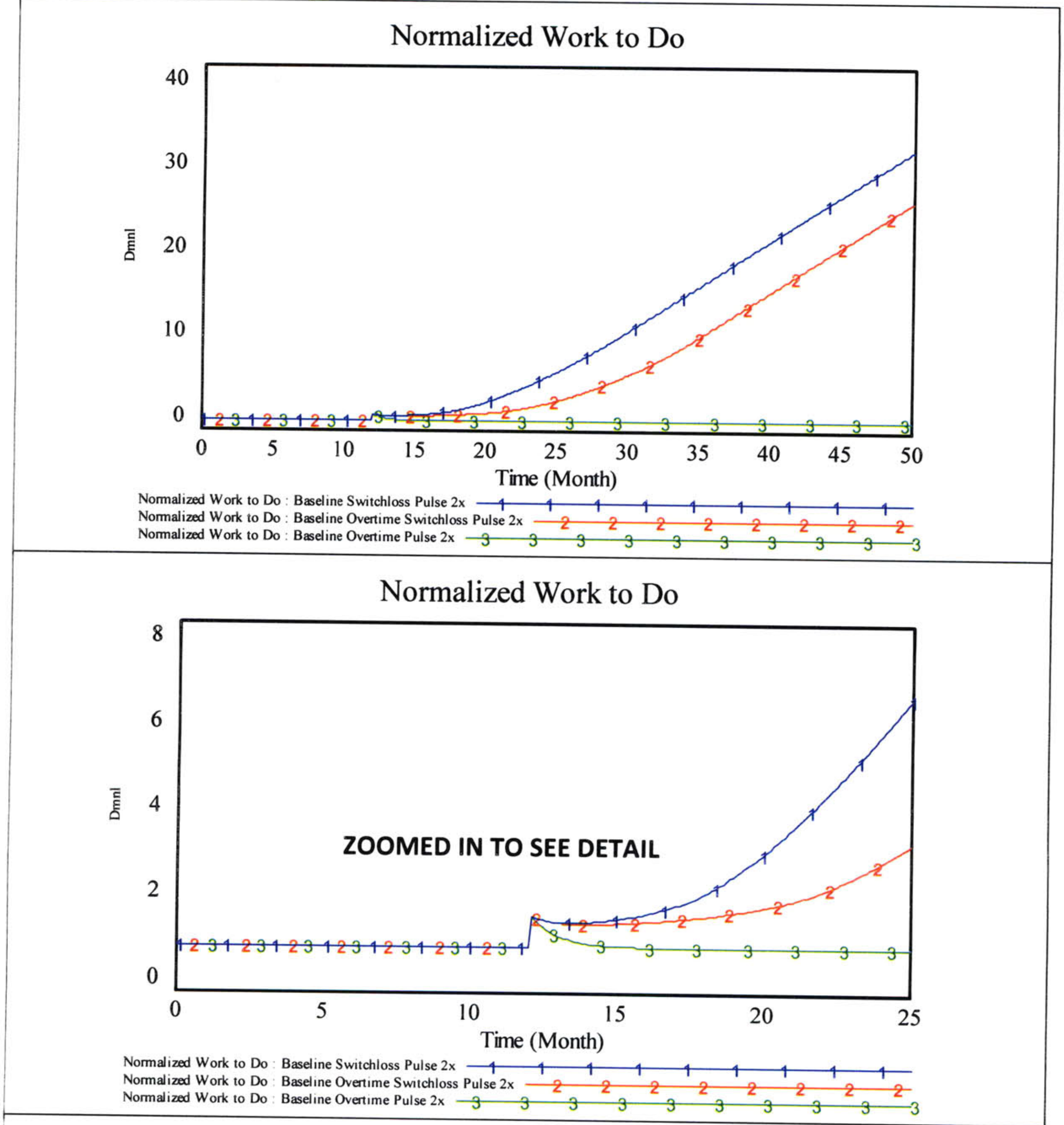


Figure 31 Multitasking - Effect of Increasing Work

The effect of switchloss is amplified with more work. The larger pulse took much longer to complete.

With Switchloss activated, there is a tipping point at which productivity drops so low that Work to Do grows exponentially.


Normalized Work to Do

Normalized Work to Do

ZOOMED IN TO SEE DETAIL

Figure 32: Multitasking - Effect of Overtime

Though the Baseline scenario recovers from the pulse [3], the Switchloss runs do not. Overtime delays the exponential effects of switchloss for a little while where growth is not as steep [2]. But once Overtime hits the limit, switchloss becomes exponential. Without Overtime [1], the effects of switchloss are more immediate.

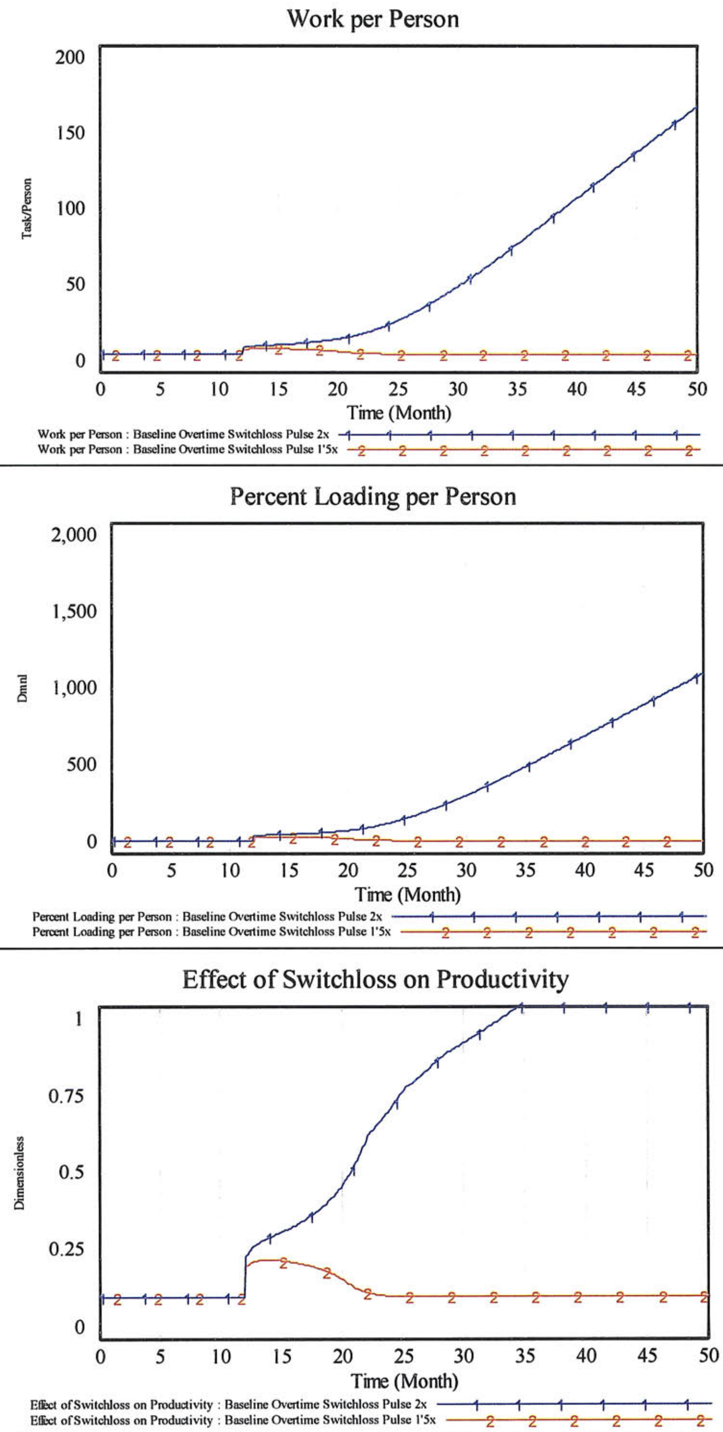


Figure 33: Multitasking - Effect of Pulse Input on Work per Person, Percent Loading, and Switchloss

To understand the tipping point, the 1.5x pulse that did not tip is compared to the 2x pulse which did tip. Work per Person, Percent Loading, and Switchloss continually increase in 2x.

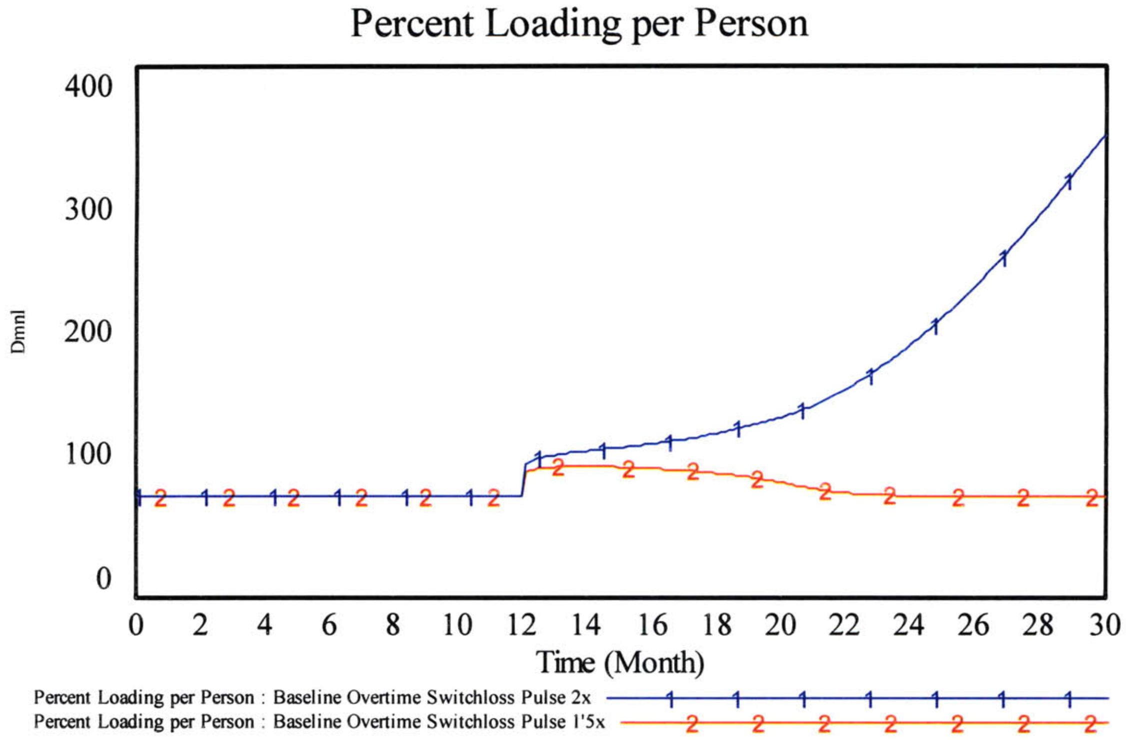


Figure 34: Multitasking - Percent Loading per Person

Zooming into get a closer look, the start is when Loading goes above 100%. Given the structure and parameters, every little bit of work above 100% loading, drops productivity just a little bit below the ability to keep up with the ongoing inflow of work. This means there is more work that is not getting done which drops productivity even further.

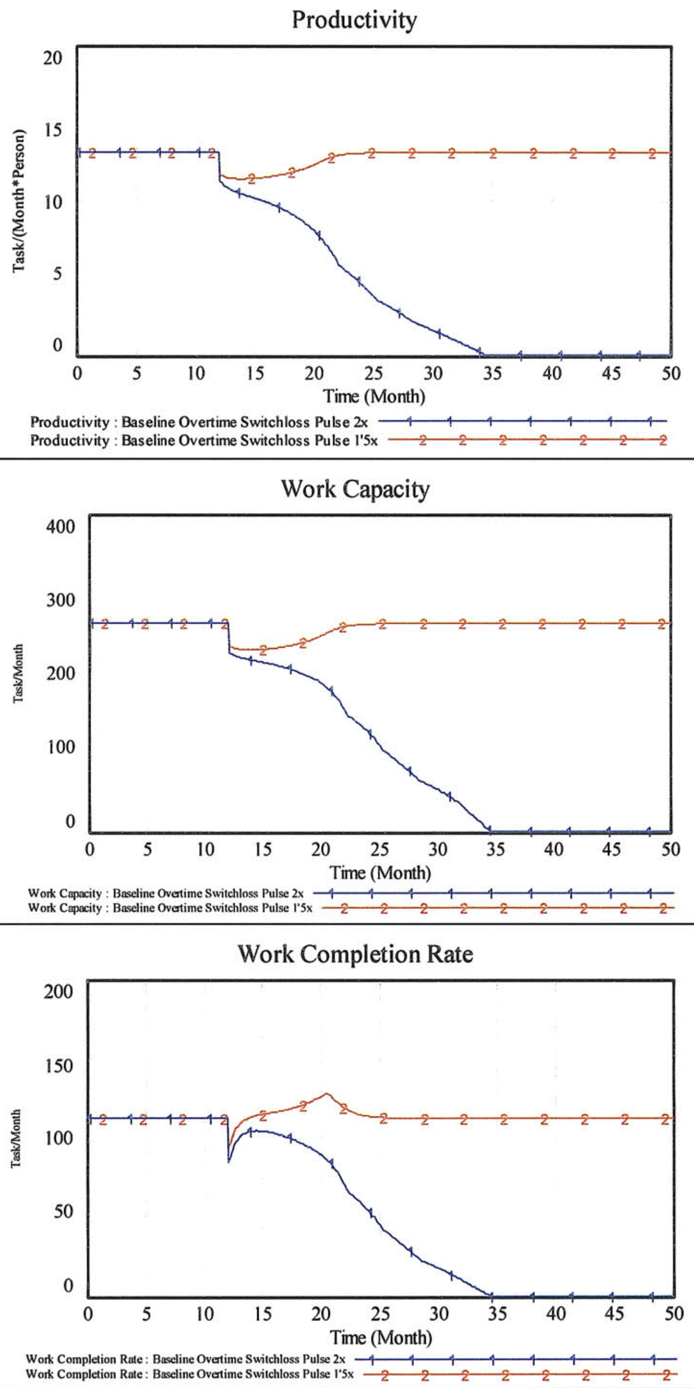


Figure 35: Multitasking - Effect of Pulse Input on Productivity, Capacity, and Work Completion Rate

At the trigger point described in Figure 34, Productivity and Work Capacity drop. Once Work Completion Rate drops below the incoming rate, the system can never catch up. It just gets worse and worse. Once it is triggered, it is a downward spiral.

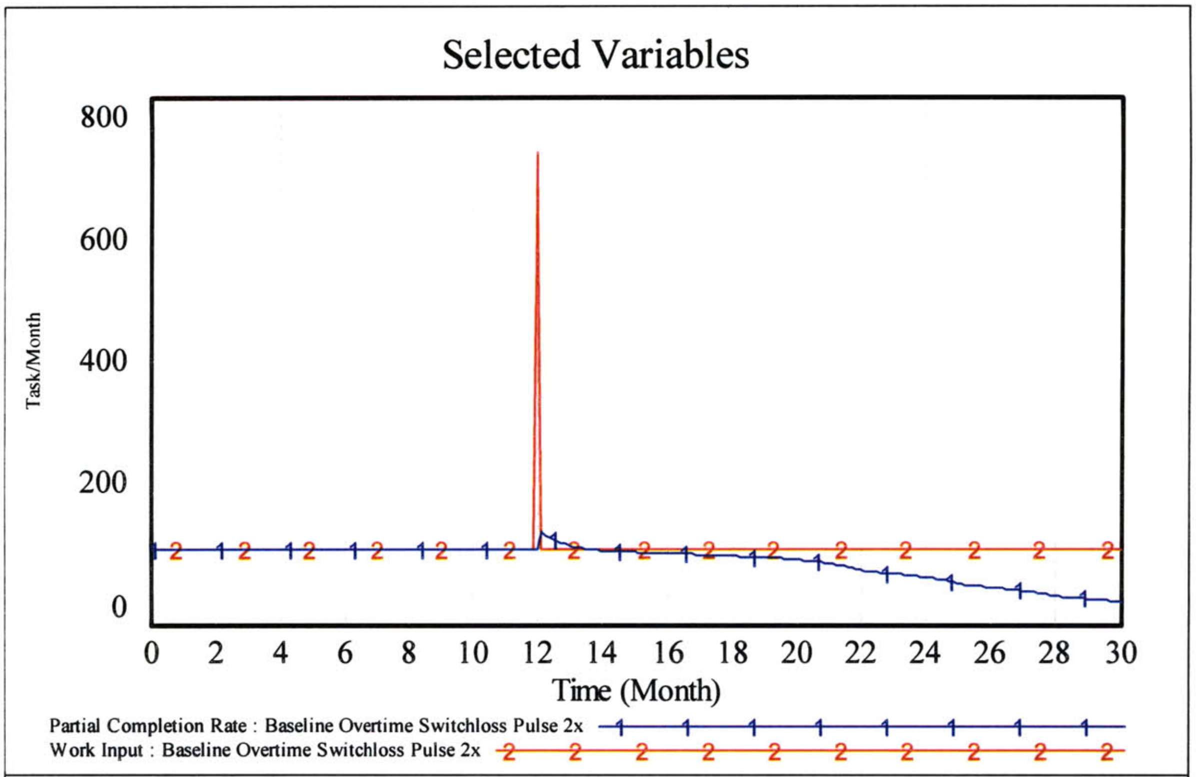


Figure 36: Multitasking – Tipping Point

At about Time=14, the Partial Completion Rate drops below Work Input. This is the tipping point at which the system does not recover and Work to Do grows exponentially. It is imperceptible at first, but compounds quickly.

Even with a one time pulse increase in workload, productivity decreases due to the increased switchloss and the system never recovers because productivity keeps deteriorating to a level below incoming rates. The one time overload sends the entire system spiraling downward.

4.2.4 Reinventing the Wheel Loop

Basis

Product Engineering has high turnover. The complaints about turnover are well founded. It is a clearinghouse for the college hire rotation program, temporary assignments, and the training ground for anyone who wants to get exposure to the product. These assignments can last for as long as 2 years, or be as short as 3 months. At one point, up to 30% of the department was in the college hire rotation program in which people rotate in and out of the department every 6 months. Although a great benefit to the company and individuals in the program, this is a tremendous burden on the permanent employees in the department and makes it a challenge to get the work done while providing a rewarding experience for the rotatees. Colleagues in other engineering groups have complained for years that they constantly have to bring new PE's up to speed on programs.

In project models, new hires are typically modeled with a negative effect on Productivity for some time as they gain experience on the job and become fully productive. They also have a negative effect on the productivity of the whole department as veterans need to take time away from their work to mentor and coach new hires and help them learn the ropes. Because these effects are well documented in the literature, it is not replicated here.

Instead, this model is intended to capture the fact that every time someone leaves, their partially complete workload gets redistributed to others in the department and a portion of that work needs to be reviewed or redone completely. Unfortunately, the more work each person generates, the more the next person must understand before starting. For example, someone leaves in the middle of a project that requires testing. Best case, he passes on the test plan that was written. The new person may not need to rewrite the whole thing, but will need to spend time reading it and any reference documents associated with it before he can conduct the test. Worst case, the test plan is lost on a hard drive unbeknownst to the new person, and is rewritten from scratch. All of this is not rework of poor quality as described in the Specialization model, but Reinventing work that was lost in the transition from person to person.

Structure

A new flow, Reinventing is added to the baseline structure. With each turnover, it takes work out of the Work Partially Complete stock and adds it to the Work to Do stock. It is assumed that all work is equally distributed across the department at all times; hence, if 10% of people turnover each month, then 10% of the total work is redistributed, and of that 10%, a fraction is reinvented.

The Reinventing loop runs clockwise. Starting with Turnover, as it increases, Reinventing increases, Work to Do increases, Partial Completion Rate increases, Work Partially Completed increases, and Reinventing increases again. Because Reinventing increases with every pass, this is a reinforcing loop that builds upon itself.

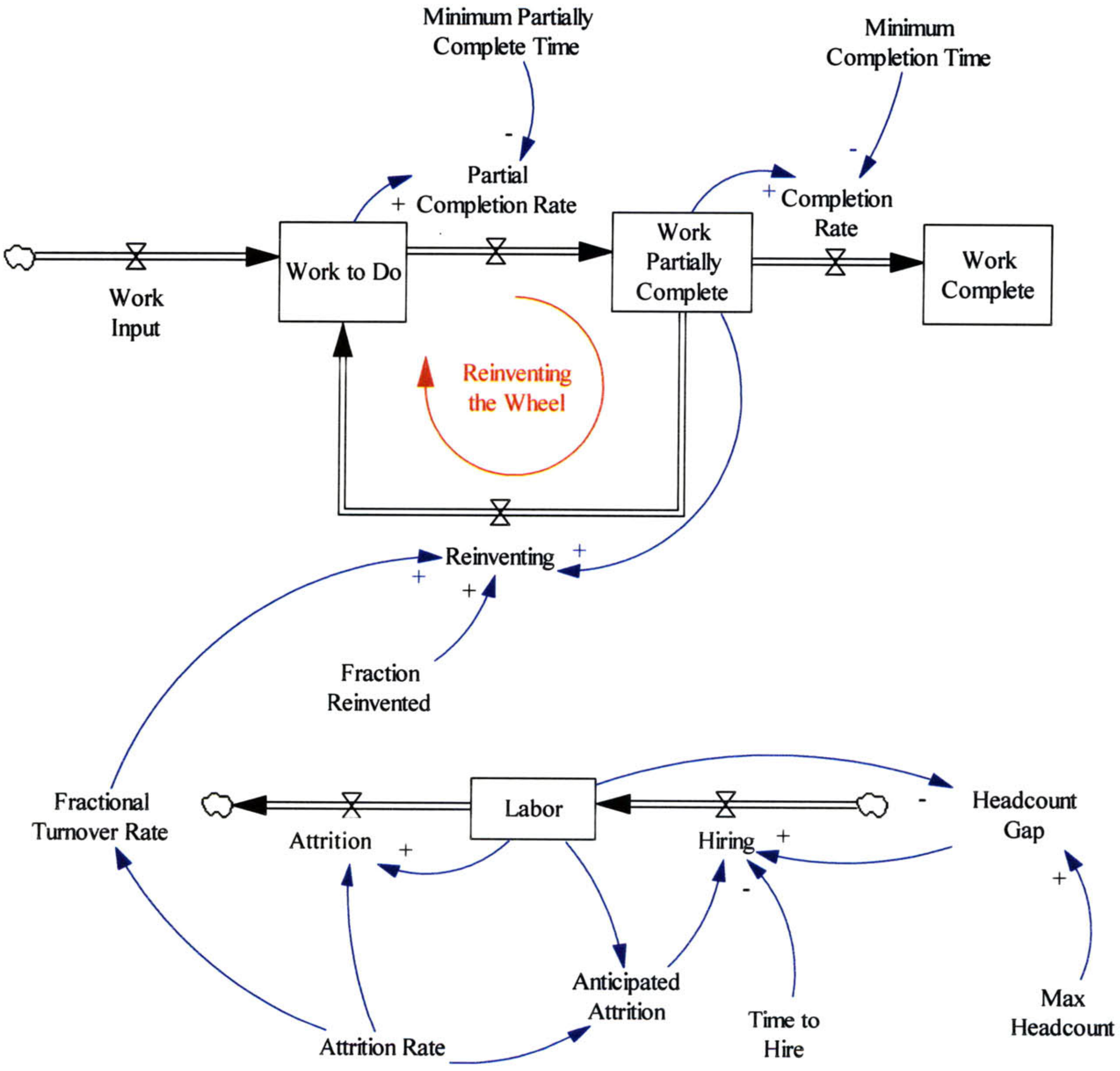


Figure 37: Basic Reinventing Structure

Key Variables
Only key variables are described in detail in the thesis. For all others, see Appendix B.

Fraction Reinvented is the fraction of work that is repeated when a project is handed off from one person to the next. When people leave an organization, they take their knowledge with them. Unless their work was documented 100% (which is impossible), there will be some knowledge lost and some amount of work that must be repeated when the next person takes over. If the person is still in the company, less knowledge will be lost because they can still be reached to answer questions or to do a knowledge transfer with the replacement. However, if they leave the company all together, none of their knowledge can be retrieved. There are also simple tasks such as reading test reports and digging up historical data that need to be repeated as the new person gets up to speed. Before the person can start contributing, they must go through and understand what others have already done before them.

Reinventing increases Work to Do at a rate that is a Fraction Reinvented of Work Partially Complete multiplied by Fractional Turnover.

Attrition decreases the Labor stock when people leave the department.

Attrition Rate is the percentage of labor that leaves the company each month.

Hiring increases the Labor stock at a rate depending on the Headcount Gap and Time to Hire.

Time to Hire is a delay that accounts for the time it takes to post a position, filter resumes, conduct interviews, and find qualified personnel.

Max Headcount is an exogenous variable because it is usually set by the larger organization. The department manager does not have direct control over it.

Rotation Rate is the percentage that goes on rotation each month. In order to keep Labor stock steady so it does not confound the effects on Work Capacity, it is assumed that as one person rotates in, another goes out.

Fractional Turnover Rate is sum of Attrition Rate and Rotation Rate.

Anticipated Attrition based on an Initial Attrition is taken from experience and recent history. It was added because not anticipating turnover leads to a lower equilibrium state due to the delay from Time to Hire. This is not just an artifact of the model. It has a basis in real life when turnover is faster than the ability to hire, there is a constant shortage of labor. The hiring process takes time (posting a position, collecting/reviewing resumes, interviewing, accepting, and relocating), so if too many people leave, it is difficult to replace them fast enough.

Behavior Analysis

Testing Scenario: Pulse Input

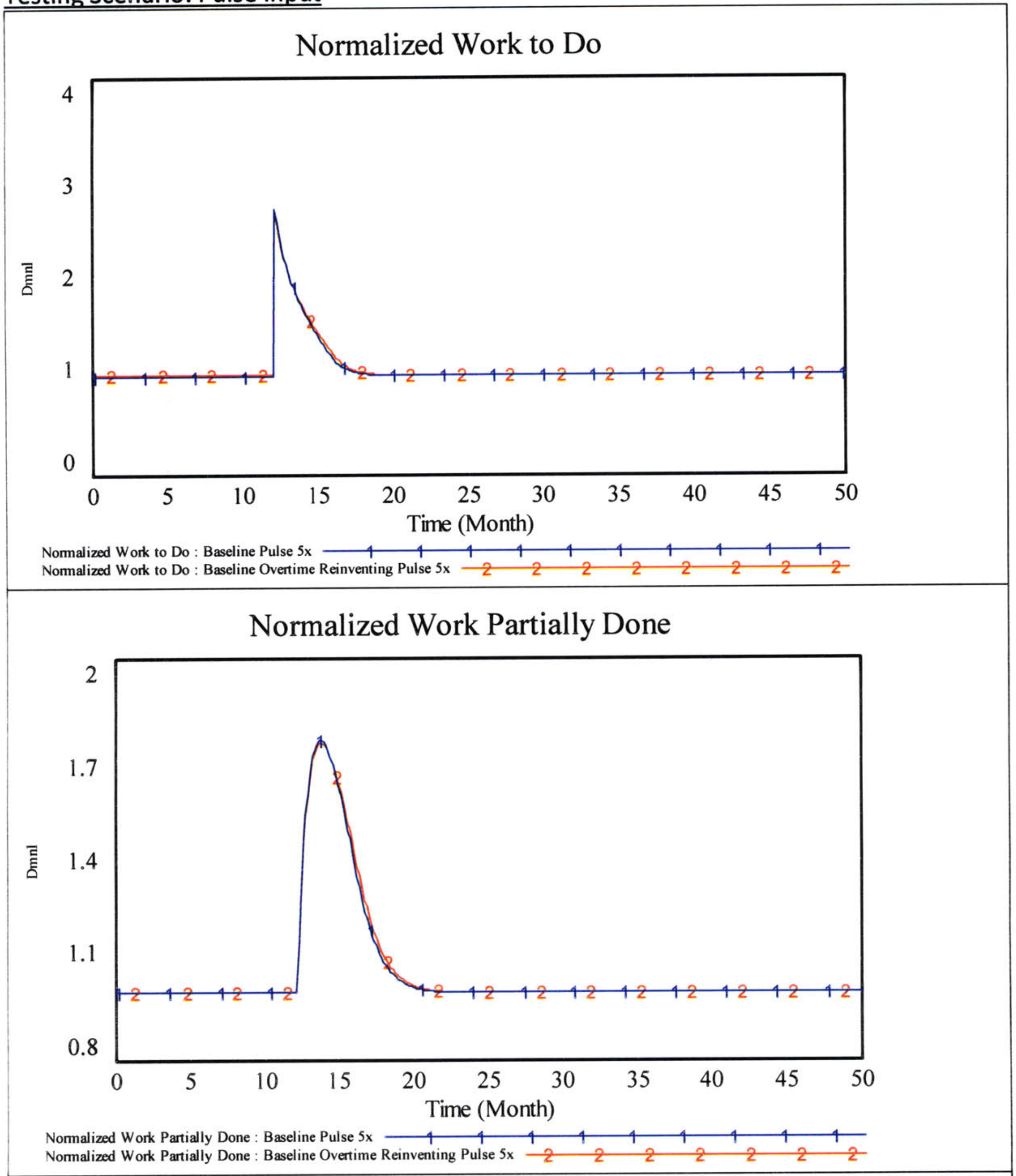


Figure 38: Reinventing - Effect on Pulse Input

Activating Reinventing alone does not have a great effect though there is a slight delay in completion. The 5x run was chosen just to show a visual difference.

4.2.5 Combining Reinventing and Multitasking and Overtime

Reinventing alone only adds a small amount of work to the pile Figure 38. However, the delayed completion from switchloss amplifies the effect (Figure 39).

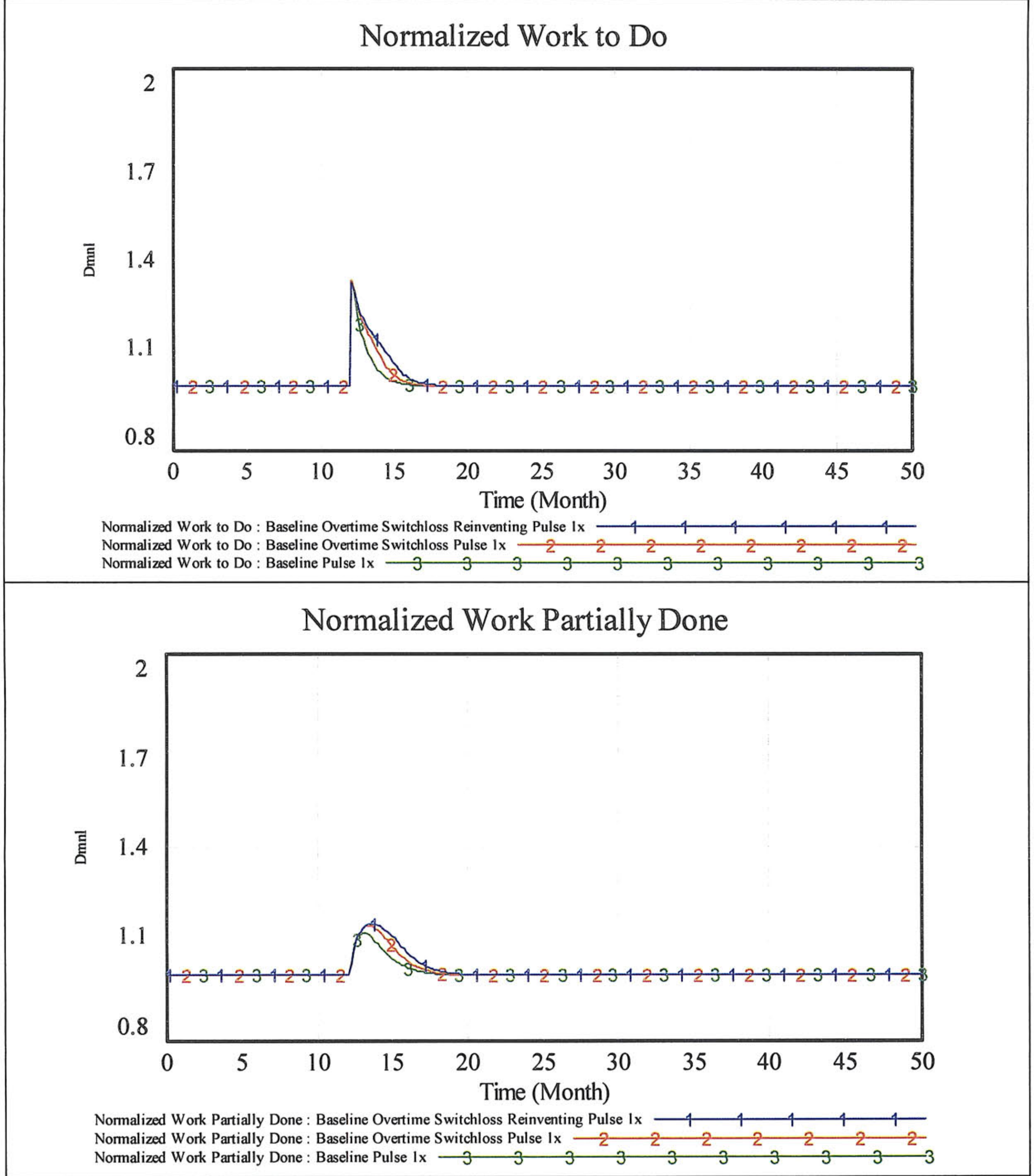


Figure 39: All Activated - Effect on Pulse Input

With Multitasking turned on as well, Reinventing has a larger effect. More work is generated and takes longer to finish.

Structure

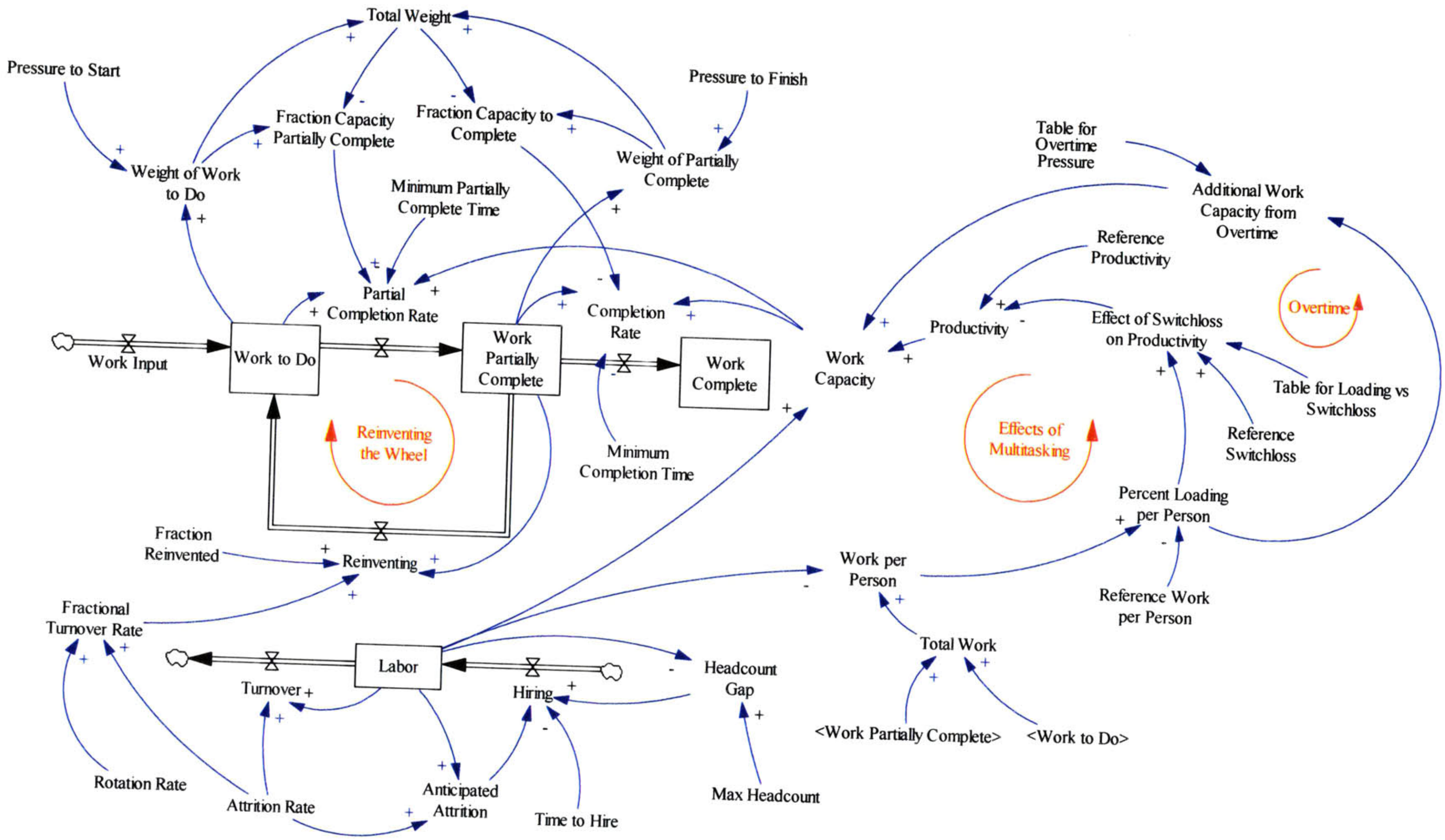


Figure 40: Final Reinventing the Wheel Structure

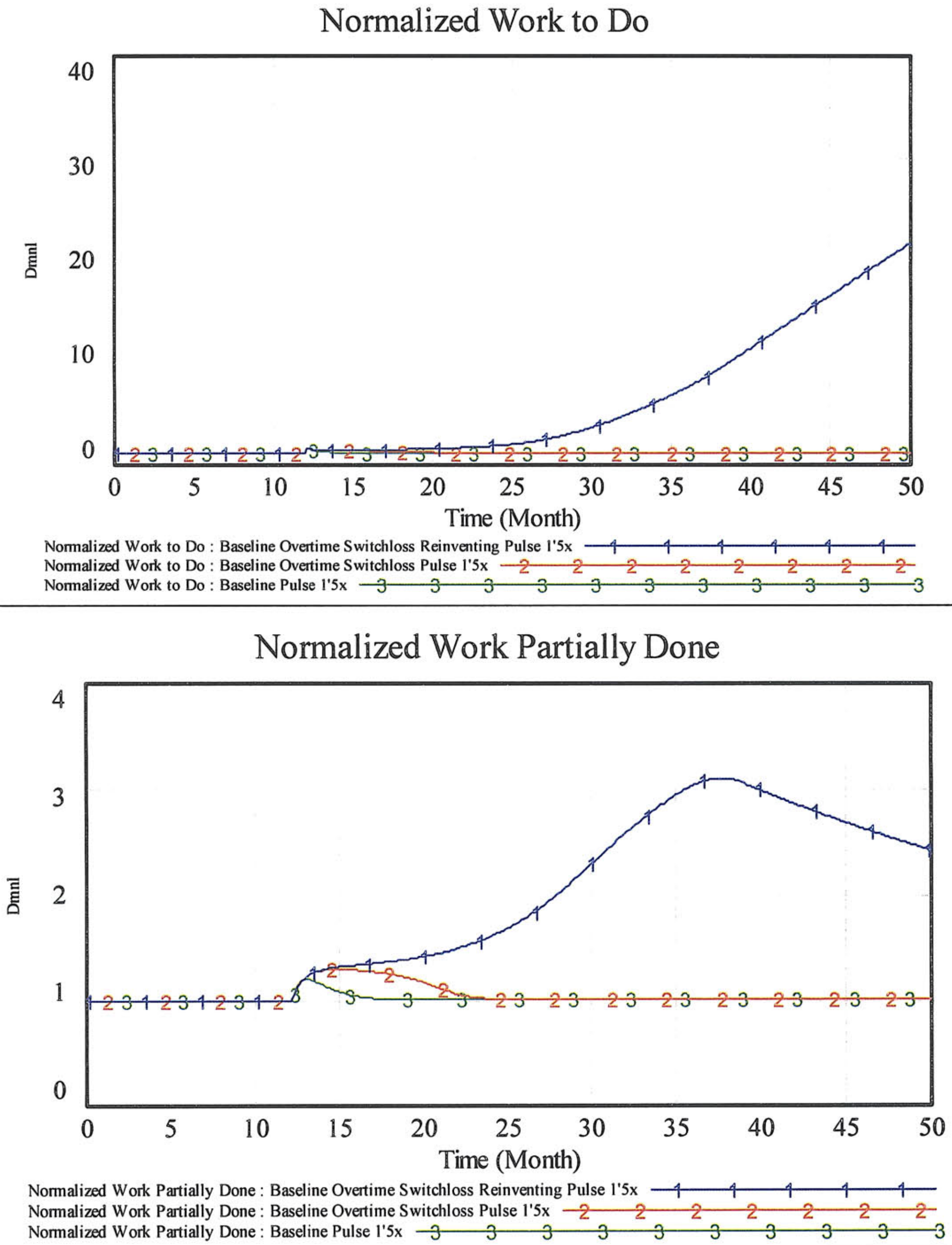


Figure 41: All - Effect of Increased Work

The effect of Reinventing is amplified with a larger amount of work. With Reinventing turned on, the extra work is enough to put the system past the tipping point and trigger the Switchloss spiral.

Policy Testing: Rotation Program

Rotations increase turnover in the department without actually lowering headcount.

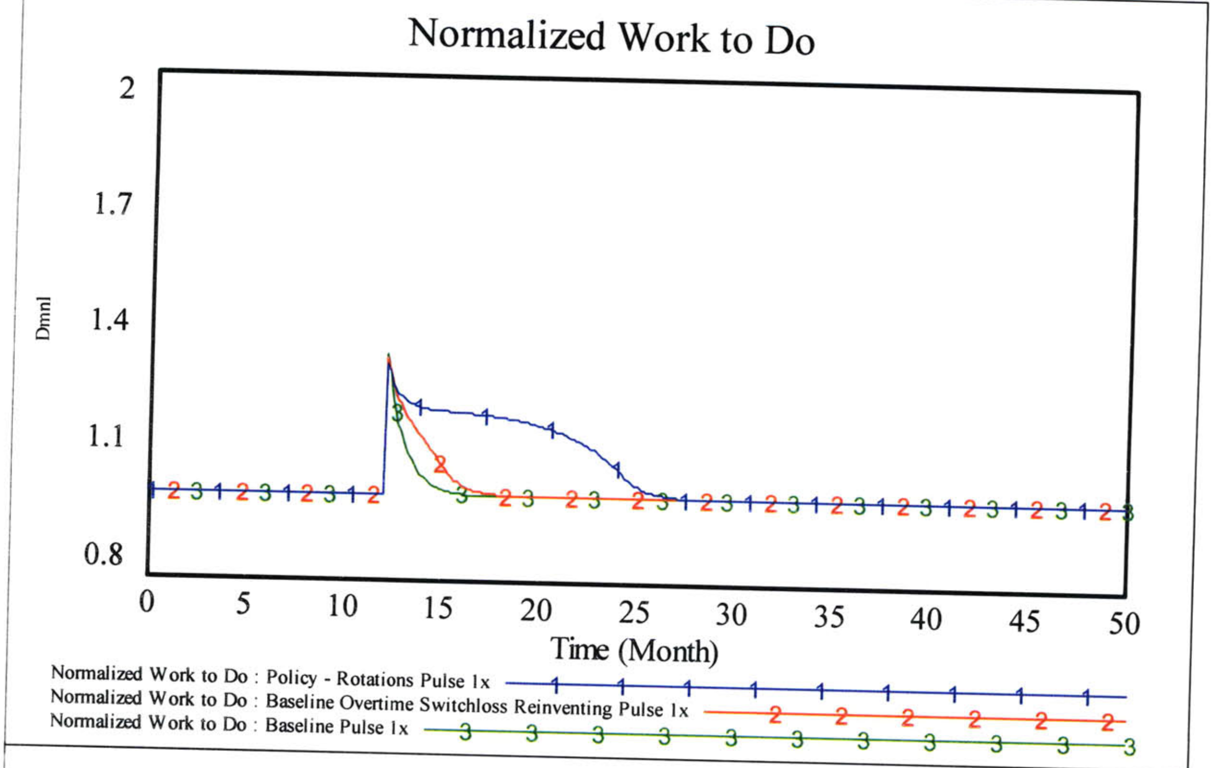


Figure 42: Policy Testing- Effect of Rotations

With all loops turned on, Rotations create additional work and it takes twice as long to recover from the pulse input.

Policy Testing: Emphasizing starting work but not finishing.

Instead of allowing the size of work stock dictate resource allocation and what gets done. Managerial pressure is added to emphasize either start or finish to influence the outcome. At Time=12, additional weight is put on starting work by allocating more resources to Partial Completion Rate.

Basis

Current managerial policy emphasizes work at the beginning, but no pressure at the end. Politically, it is important to show progress on work, but not necessarily complete it. People ask if anyone is assigned to the project, but do not follow up to ask if it was ever completed. Focus and attention is very high at the beginning of projects and through launch, but wanes toward the end. Finalizing documentation and cleaning up specifications is not as glamorous and loses visibility within the organization.

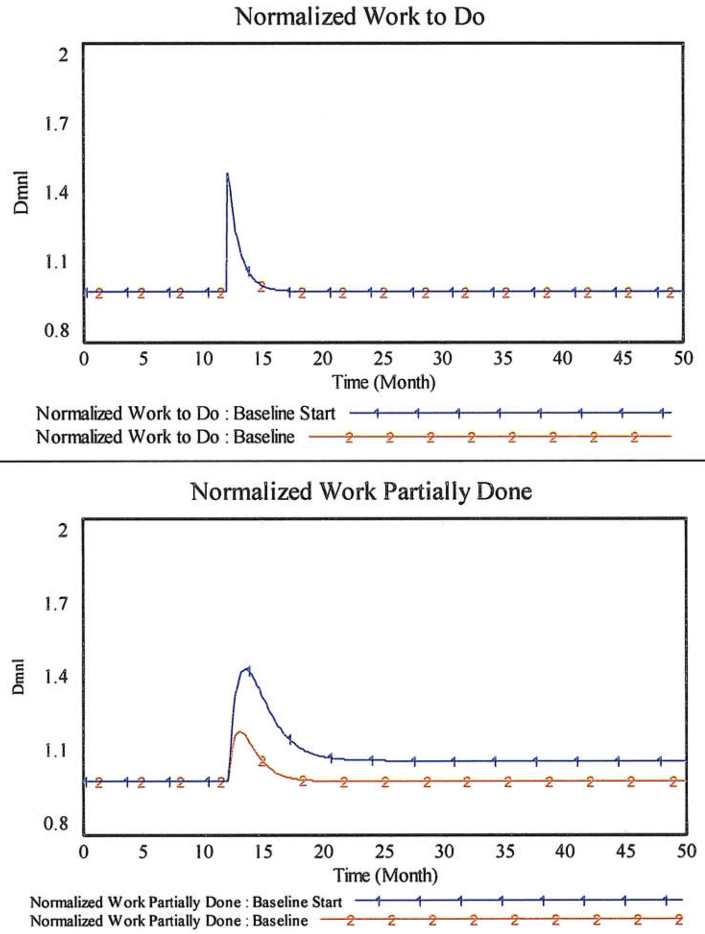


Figure 43: Policy Testing- Effect of Pressure to Start on Baseline Scenario

For these simulations, Pressure to Start=1 which allocated more resources to start Work to Do. Because the initial loading is low, Work to Do is complete at the minimum possible time even though more capacity is allocated. Because resources are diverted to start Work to Do, fewer resources are available to complete the Work Partially Done. So, there is a step change in Work Partially Done which means there is always more work than capacity to complete.

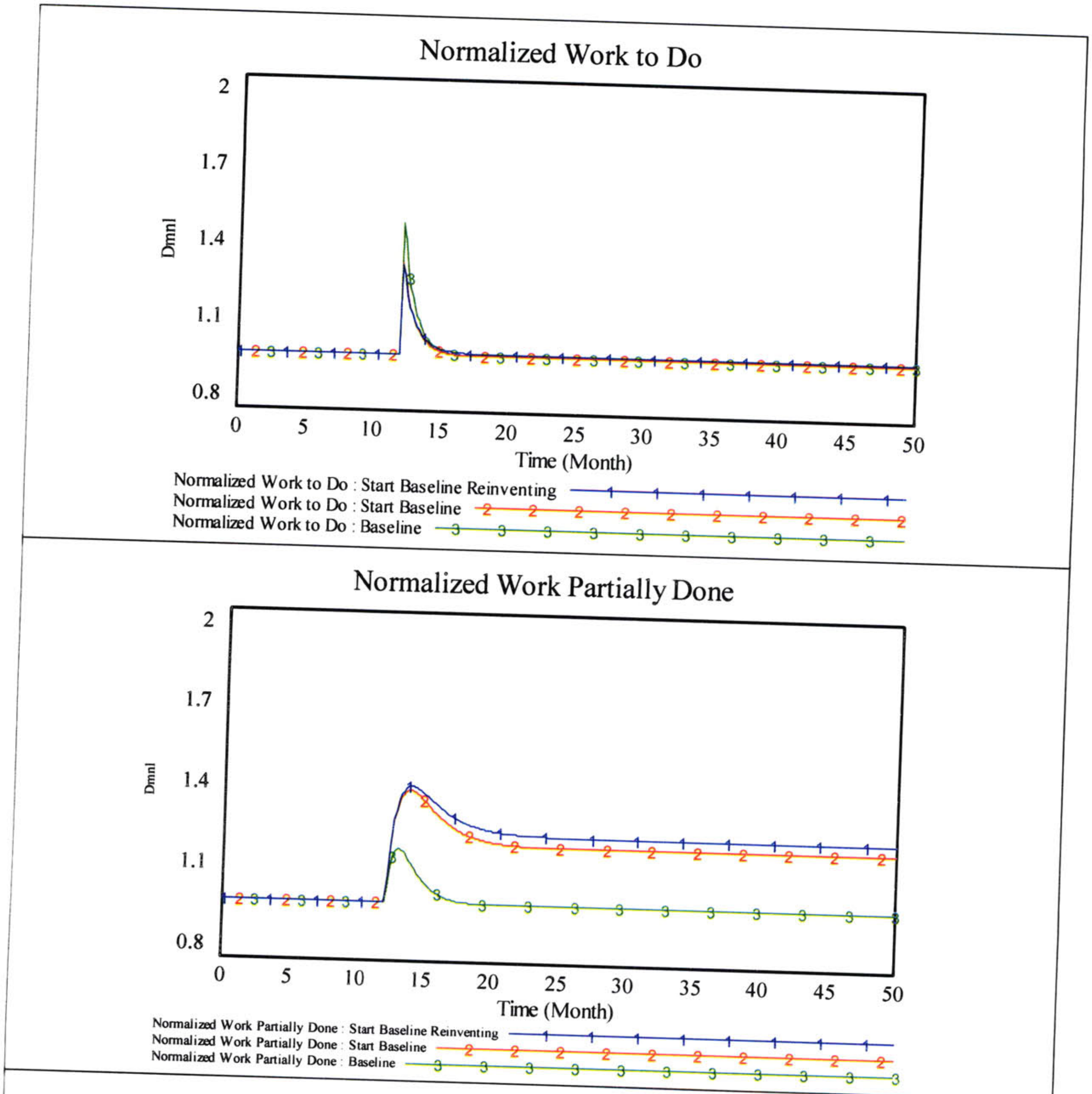


Figure 44: Policy Testing - Effect of Pressure to Start with Reinventing

Similar to its effect on Baseline, Pressure to Start decreases Work to Do faster, but causes Work Partially Done to never recover.

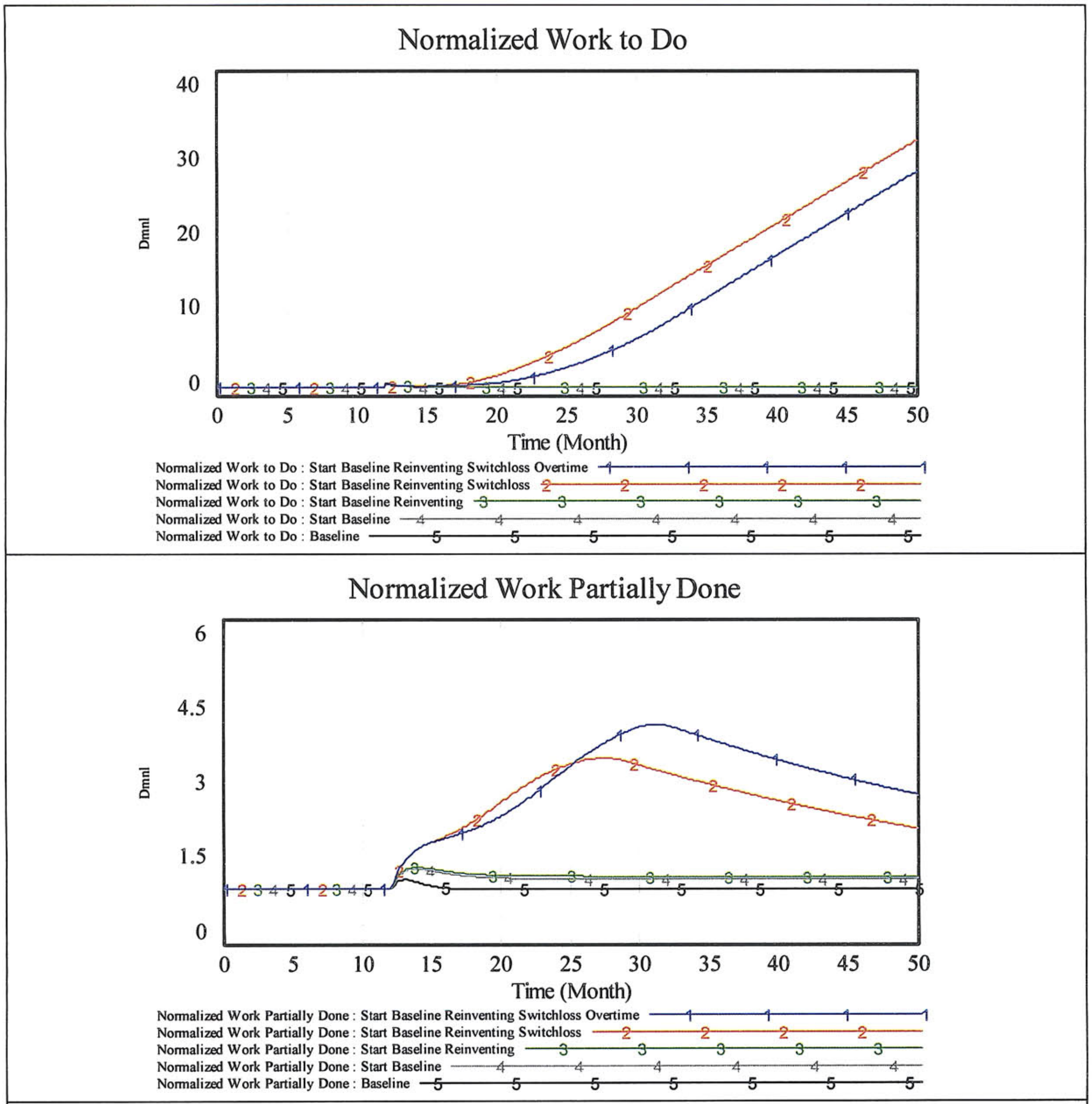


Figure 45: Policy Testing - Effect of Pressure to Start Policy with All Activated

With all loops activated, Pressure to Start focuses all the resources on Work to Do which looks great from T=12-20. In the meanwhile, Work Partially Done increases enough to trigger Switchloss and Productivity plummets for everyone from T=20 on.

Work Partially Done declines after T=30. If tasks take too long to complete, it will negate all work done to date and the work is sent back to Work to Do for the next person.

Policy Testing: Pressure to Finish

At Time=12, additional weight is put on starting work by allocating more resources to Completion Rate.

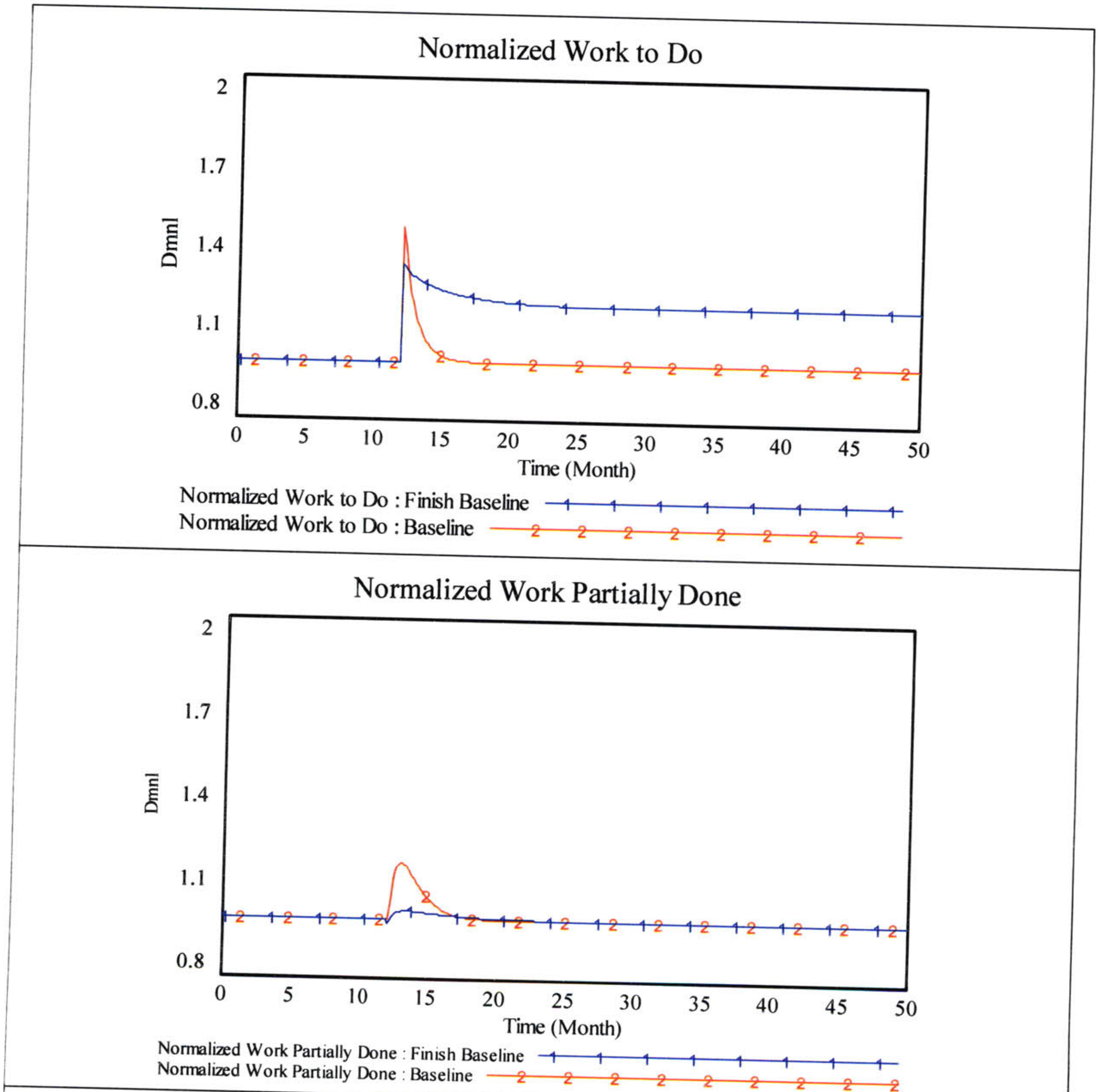


Figure 46: Policy Testing - Effect of Pressure to Finish on Baseline

Putting on Pressure to Finish reduces the Partial Work, but creates a step change in Work to Do.

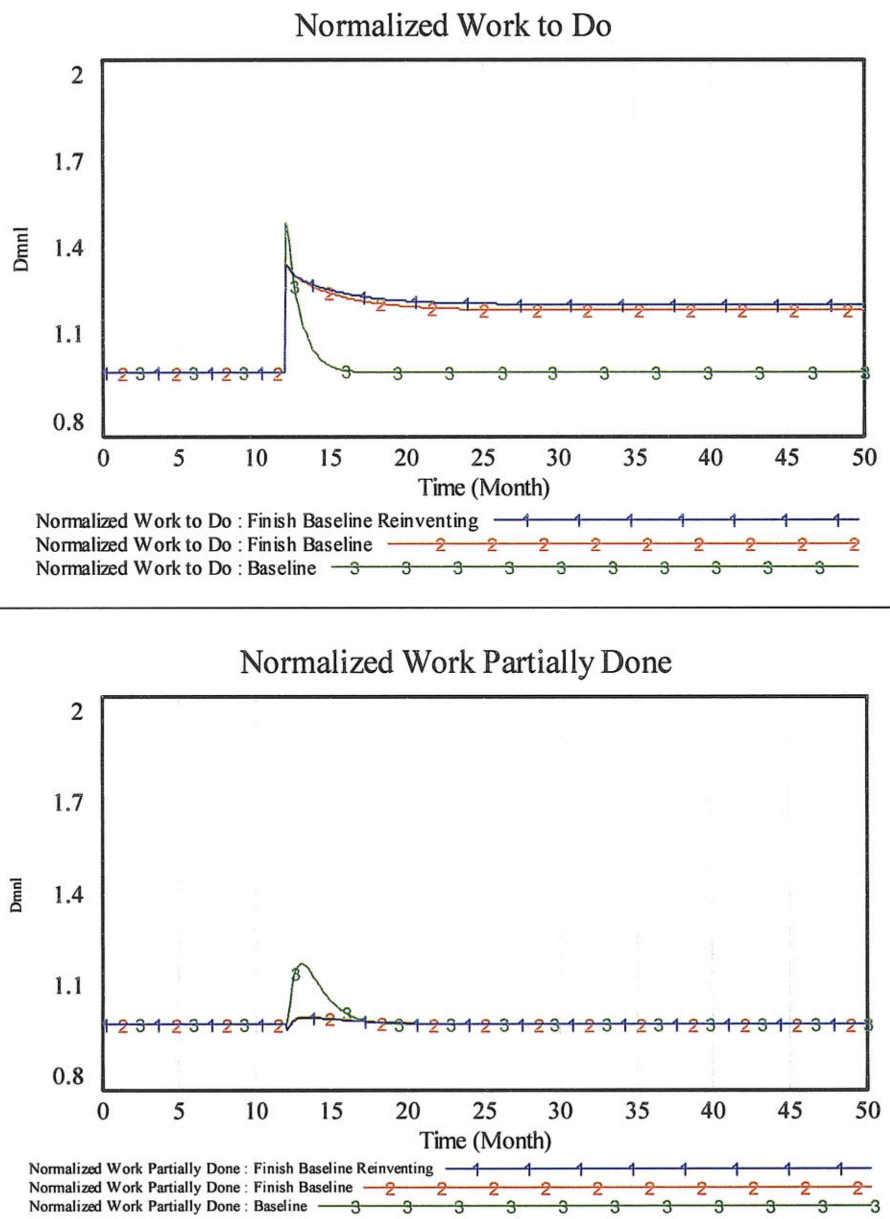


Figure 47: Policy Testing - Effect of Pressure to Finish with Reinventing

Turning on Reinventing, there is no significant difference to Finish Baseline.

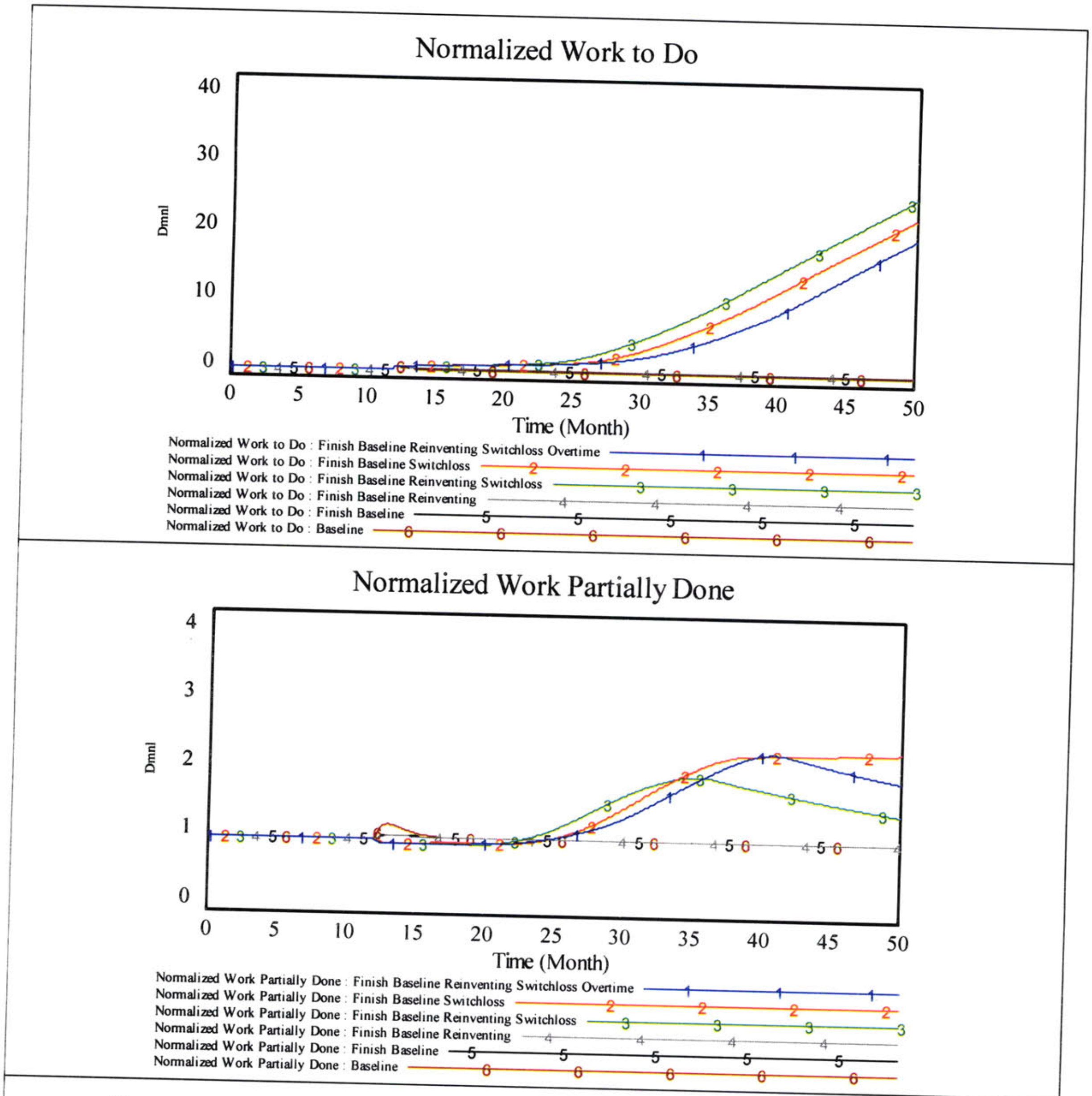


Figure 48: Policy Testing - Effect of Pressure to Finish with All Activated

With the Switchloss on, Partial Work is very low for a while, but at the neglect of Work to Do. It looks good for a while between Time=12-24, but eventually, the Work to Do piles up and productivity declines toward zero. Work Partially Done declines toward the end [1, 3] because reinventing is putting that work back into Work to Do. Work Partially Done does not continue to increase because eventually, no work is flowing in from Work to Do.

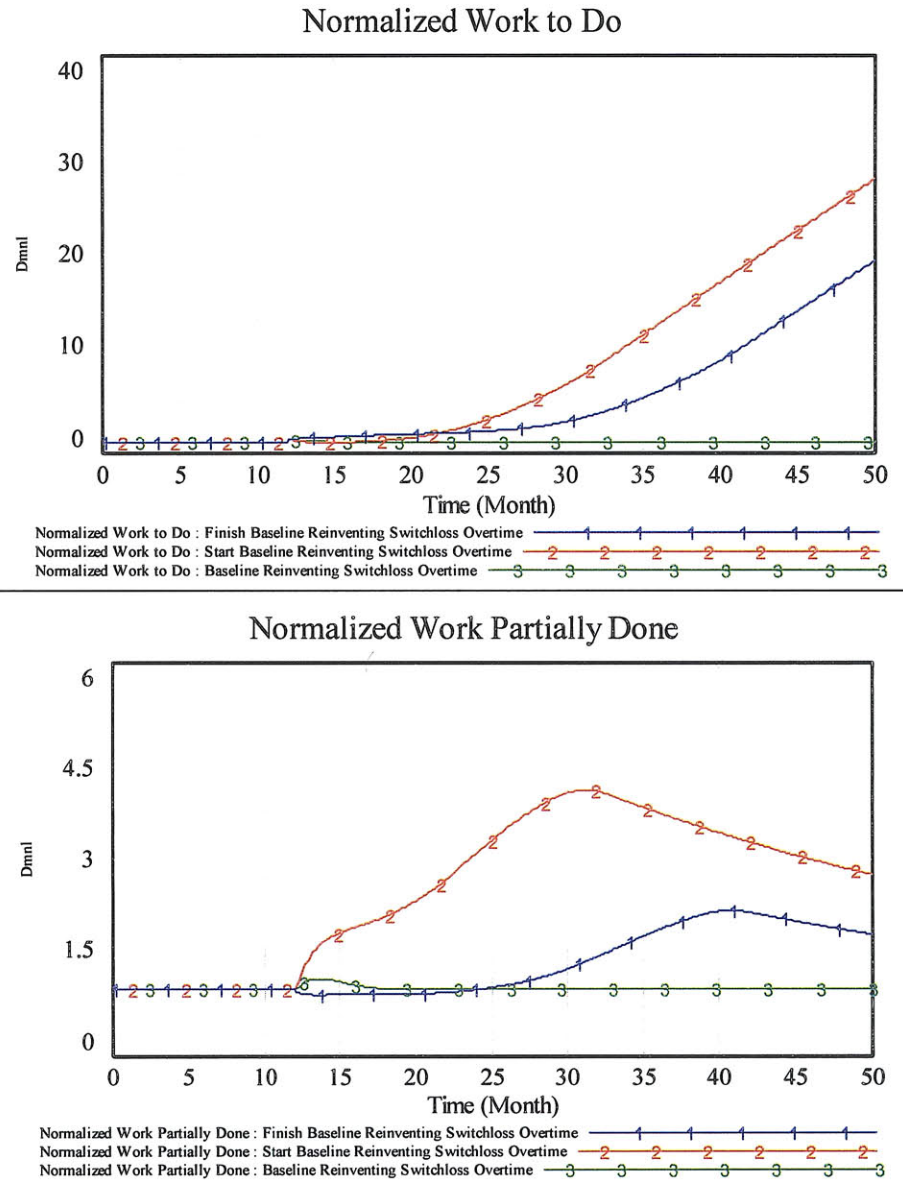


Figure 49: Comparison of Resource Allocation Policy

An equal emphasis to start and to finish work is best [3]. Overallocating one versus the other leads to overload. Letting work pile up in either stock triggers the cascading switchloss throughout the system. Putting pressure on to finish projects [1] is more effective than to start projects [2]. However, it only works for a period of time before Work to Do piles up too much.

As shown above, the impact of both Switchloss and Reinventing can be significant depending on initial circumstances. The effects of Reinventing is often not recognized or planned for in project management. Industries or departments with high turnover and programs that take longer than the expected tenure of resources should pay attention to turnover to manage workload.

Existing workload and switchloss effects are often not considered during portfolio planning. Depending on how loaded everyone is to begin with, adding just one more program on top could have detrimental effects on all projects.

Current managerial policy emphasizes distributing and starting projects, but there is no one following up and putting pressure on people to close out projects. It is normal for people to be pulled from the end of projects to start the next latest and greatest. As a result, the old Partial Work Done keeps piling up. Worse, due to turnover, knowledge is lost and work is reinvented over and over again which just adds to the existing workload. The growing pile of old work would not be an issue except that there is enough organizational pressure to assign them to someone and keep them on someone's plate just to say that someone is working on it. However, just because it is assigned does not mean significant effort is put in. If by chance the person does get the time to look at it, they will not complete it in one pass, will put it aside for another month, and then need to review what was done a month ago before starting to work on it again. As shown in Figure 49 the best policy is to keep an even allocation to start and to finish projects.

(This page intentionally left blank.)

5 Future Work

As with any research project, finding the answer to one question leads to the discovery of ten other questions that could be pursued. This thesis is no exception. Unfortunately, a line must be drawn at some point since every project has a due date and the Overtime Loop has already kicked in to complete this thesis. In conducting the literature search, creating the model, and analyzing the model, I have found many other factors that could be added in future work. Below is a summary of each and why they could be important to explore.

5.1 Additions to Reinventing the Wheel

Repercussions of Overtime

- Overtime is an effective tool to manage fluctuation in workload if used in small doses. After a period of time, however, fatigue sets in and decreases overall productivity. Additionally, if the workload never decreases, people will stop putting in overtime because it does not seem to make any difference. Adding repercussions would trigger the Multitasking loop earlier since Overtime currently delays it.

5.2 Additions to Specialization

Time Crunch

- Specialization increases "hold-up" problems, delay of passing work from one group to another (Becker and Murphy, 1992). This would increase the time crunch at the end of any given project. This is because those at the front end have no delivery pressure. At the beginning of any program, it is usually easy to justify delay after delay because the deadline seems so far away. Unfortunately, this eats up all the safety stock built into the timelines and leads to a scramble at the end which may reduce the quality of work at the end.
- This could be an important unintended consequence of specialization that was not been considered.

Increased Attrition/Decreased Time to Hire

- Specialization could increase attrition due to boredom, but at the same time decrease time to hire because fewer skills are needed. At first, attrition will increase because people currently in the department may get bored with a reduced scope of work.
- If specialization increases Attrition enough, it could trigger the Reinventing loop.

Quality

- Normally, Rework is driven by Quality which is a function of Depth of Experience. It was left out in order to focus on the Breadth story. Other rework factors that affect Quality could be added so that additional tests can be conducted to understand the interactions and tradeoffs between Depth and Breadth.

Causes of Specialization

- Because specialization is an organizational design decision at this time, it was left as an exogenous decision in the model and the focus was on the effects of specialization. In the future, more work could be done to explore the causes of specialization to understand the dynamics that play into the decision to divide labor into ever smaller increments.
- Understanding the causes and effects further could lead to better understanding of how bureaucracies are generated, and how to keep a large organization from becoming a bureaucratic quagmire.

6 Conclusions

The models have illustrated some of the pitfalls of specialization and provided alternate explanations to the overload that Product Engineering is experiencing.

6.1 Insights about Specialization

Though Specialization may increase Productivity initially, the gains can decrease over time due to delayed effects of coordination cost and rework. If the new equilibrium is higher than the initial Productivity, it is worth the cost if you can afford the rework and potential delays in product delivery.

Typically, it is easy to track Work Completion because the deliverables are highly visible within any project. Often, what gets overlooked is the amount of rework that is generated during a project. Unless the rework is a huge catastrophe or has astronomical costs, no one sees it. It is buried as part of the process. Also, the long delays associated with the rework means that the root cause may never be identified. By the time rework is discovered, it is usually at critical junctures within a project, and everyone scrambles to fix it and move forward. Rarely is there time to ponder about why it occurred in the first place.

Breadth of Experience affects the quality of interdepartmental handoffs and reduces downstream rework, whereas Depth of Experience affects the quality of intradepartmental rework. Specialization increases one while decreasing the other. Specialization makes sense when there is little interaction and reliance from one department to another, but it may not make sense when there is a complex interdependency of activities. There is such a thing as too specialized.

6.2 Insights about Multitasking

When that extra project sends the system into overload conditions, there is no way to recover. It makes it worse not only for that one project, but for the entire system as well. Multitasking is a downward spiral where switchloss breeds switchloss. The model presented here is an aggregate representation of the product development process where multitasking effects overall productivity of one department. Yaghootkar and Roos (2007) show that multitasking across too many projects can deteriorate performance of the entire product portfolio.

6.3 Insights About Overtime

The insight from testing this scenario is that organizational pressure to put in overtime is needed to complete projects. Given status quo and no pressure, additional work just gets piled on top, something falls to the bottom of the pile and the low priority work will never get done. If management is not asking for it, then it is assumed to be unimportant and will not be worked on. Management can increase pressure to finish the work even faster.

Overtime is needed to offer some flexibility of workload. However, constant overload is detrimental to an organization. There is only so much overtime that can be utilized to reduce the overload. The original rework model shows that fatigue will be triggered after sustained overtime and reduce productivity after a period of time. Overtime should be used sparingly but can be an effective managerial tool.

6.4 Insights from Reinventing the Wheel

Both multitasking and turnover can explain why things seem to take forever to get done, if at all. Multitasking exacerbates reinventing the wheel. The longer it takes to finish, the more work is generated due to turnover.

Even with a constant workload without overload conditions, in a job where historical knowledge is important but projects have long completion times, assuming constant tenure/turnover, each person has less time to work on a project because they have to spend more time learning what others have done in the past.

6.5 Conclusions

The biggest insight for me through this process was the realization that much of the work is self-generated. I thought of turnover as a hit to work capacity, but not as a factor that actually generates additional work. This led to the identification of other factors that contribute to reinventing the wheel such as the reshuffling of work and multitasking.

Specialization could be the best solution to overload, but it depends on the reason for overload. If the reason for overload is rework, reinventing, or turnover, other solutions may yield the same productivity improvements without the coordination cost or additional rework.

In the case of Product Engineering, there is a high degree of interaction between knowledge gained during the design phase and implementation into manufacturing. Splitting the department into two would sever the learning and could have detrimental effects on the product development process over time. However, given the current overload situation, it may be necessary to specialize to some degree to offer some relief from the Productivity gains.

7 Recommendations

7.1 Specialization

- Product Engineering's strength is also its weakness. Product Engineers are valued because they bring a broad range of experience to the table during any phase of the product development process. The rotation program is a great way to accelerate the rate of breadth gain. However, this leads to high turnover and increased workload on the department. One suggestion is to reduce turnover and not have so many new college hires in the department. This will lead to less reinventing. However, this would lead to less breadth of experience and a delayed increase to rework.
- Nothing can substitute actual hands on experience, but, to some degree it is possible to learn from other's experiences. If interdepartmental knowledge sharing mechanisms like project reviews were put in place and consistently run/attended, some breadth of experience can be gained. Unfortunately, under overload conditions, training goes to the bottom of the priority list because no one is clamoring for it to be done today. As workload increases, people buckle down and just do what they need to do to stay above water. To be successful, not only must cross training be seen as an important activity by the entire organization, but time must be given to enable people to attend. Otherwise, it is just another meeting that takes time away from doing real work.
- Informal networks can be powerful mechanisms within an organization (Krackhardt, 1993). Encouraging informal networks to develop so that people share stories over lunch to increase breadth of experience. "Doing lunch" is another luxury that diminishes as an organization becomes overloaded.
- Care should be taken when redesigning the product development process and assigning responsibilities across departments to minimize the negative effects of specialization.

7.2 Reinventing the Wheel

- As a manager in charge of project assignments, reducing the number of project handoffs from person to person is an important step in reducing additional work.
- Anticipating Turnover will make the system more resilient to unanticipated turnover and maintain a stable headcount. A 6 month delay in hiring can be significant for an already overloaded department.
- Reinventing could be reduced through better documentation. Managerial emphasis on proper documentation is needed for people to do it. Documentation takes away from productivity today, so it is easy for people to rationalize not doing it. If no one is asking for it today, no one will do it for future benefit. It should be noted that this policy has limited impact because only explicit knowledge can be written down. Tacit knowledge can only be transferred from person to person.

- Policies could be put in place to help employees better manage their workload and reduce switchloss. Oftentimes, all that is needed is a quiet room away from interruptions. Flexible work arrangements could help, where employees work away from the office once every week or two to get through the time intensive tasks. Learning to cluster similar tasks together also helps reduce switchloss.
- Mark, et.al (2005) found that information workers switched tasks every 3 minutes and switched work topics every 11 minutes. Their study looked at the effects of collocation on interruptions. The benefits of collocation are based on constant communication and frequent information transfer. It can speed up the learning of new hires as more experienced colleagues can chime in to help direct them. Unfortunately, the side effect is frequent interruptions and switchlosses. Mark found a distinction between beneficial interruptions or "interactions" that match the work currently in progress and "disruptions" that are completely out of context. She suggests increasing awareness to minimize disruptive interruptions. For example, establishing cultural norms like "Do not disturb" signs.
- Overtime pressure is needed to close out projects. However, care must be taken not to have the organization in constant overload conditions. Otherwise, fatigue is triggered and productivity decreases in the long run.
- Project deliverables and processes could be designed such that completion time of work packets is less than expected tenure.

7.3 Overload

- It is vital for management to monitor the inflow of work and ensure that their workforce is not constantly overloaded. As shown in Section 4, even a one time increase in work could send the system into a downward spiral depending on how loaded people are to begin with. In order to do this effectively, resource planning tools and measures are required with validated FTE (Full Time Equivalent) estimates.
- Wheelwright and Clark (1992) write "if any one project runs into unexpected trouble, there is no slack available, and it will be necessary to take resources from other projects. This causes subsequent trouble on other projects and the effects cascade." They suggest maintaining excess resource capacity. Although headcount may not be directly at the manager's discretion, he has more influence on it than changing the product portfolio. A stronger case can be made to increase headcount with better supporting data per the previous recommendation.
- The literature on firefighting is extensive. Black and Repenning (2000) show the detrimental effects of under allocating early phases which lead to firefighting and rework downstream which means under allocating even more during the next development cycle and on and on. They suggest project reviews and project cuts early on in the design phase based on available resources in order to avoid the vicious cycle.

7.4 Overall

Current managerial policy emphasizes the start of projects, but not the completion. This has led to an increasing pile of work that has been passed on from engineer to the next, compounding the already excessive workload. First, pressure should be put on the resources to finish out all the existing projects. Figure 49 shows that it could be effective for a short time. If additional resources are not allowed, then new project work will need to be neglected for a period of time. If organizational focus can complete the work quickly, the long term rework repercussions could be limited.

7.5 Warning Signs

As a manager, it is important to recognize when your people are overloaded because they are not working at their peak potential. Or worse, they will burn out soon and their productivity will plummet. One sign of loading is how often people actually take a lunch. Doing lunch is a great way to communicate and share knowledge and learnings and build relationships. It also helps to reset and reenergize during the day. If everyone is working through lunch day in and day out, then they are at the very least level loaded and teetering on the edge. If everyone is working from home and answering emails at night because they cannot get to it during the day, then the department is overloaded. If it is just one person, then perhaps that person is not as efficient as everyone else and may need to put in the extra time to keep up. However, if it is the norm across the group, then it should be taken seriously.

The insidious nature of switchloss is such that productivity drain is incrementally small, and often unnoticeable. Also, because you are working on so many things all at once, it may feel like you are being very productive. It is not until the end of the day (week/month) that you realize nothing was actually accomplished. Something that should have taken 5 minutes took 5 hours.

It is also important to pay attention to the degree of specialization within an organization because an organization can become too specialized. The amount of time people spend in meetings instead of doing "real work" could be a measure of the over specialization. When half the day is spent on reading and responding to email, coordination cost is very high. Tracking time or cost lost due to rework is a measure of the system understanding gap within the company. Additionally, if taken to the extreme with too much division of labor, the job can become unchallenging. This can lead to high turnover due to boredom. Worse, the level of bureaucracy can increase. When no one knows what everyone else is doing, processes and paperwork are put in place to standardize and coordinate movement.

8 References

Becker, Gary S., and Murphy, Kevin M., 1992. "The Division of Labor, coordination costs, and knowledge." Quarterly Journal of Economics, Issue 4, p. 1137-1160.

Black, Laura J. and Repenning, Nelson P., 2000. "Why Firefighting is Never Enough: Preserving High-Quality Product Development." Whitepaper. http://www.mit.edu/nelsonr/www

Bohn, Roger E, 1994. "Measuring and Managing Technological Knowledge," *Sloan Management Review;* Fall 1994; 36, 1; pg. 61.

Boucekkine, Raouf, 2008. "Human Capital Accumulation and the Transition from Specialization to Multitasking," Macroeconomic Dynamics, 12, 2008, p.320-344.

Crenshaw, Dave, 2008. *The Myth of Multitasking: How "Doing It All" Gets Nothing Done.* Jossey-Bass. c.128p.

Eppinger, S.D., Whitney, Daniel E., Smith, Robert P., and Gebala, David A., 1994. "A model-based method for organizing tasks in product development," Research in Engineering Design 6:1-13.

Ford, David N. and Sterman, John D., 2003. "Overcoming the 90% Syndrome: Iteration Management in Concurrent Development Projects", *Concurrent Engineering: Research and Applications*, 11(3): 177-186.

Greenberg, Jerald and Baron, Robert, 2003. *Behavior in Organizations*. 8th ed. Upper Saddle River, NJ: Prentice Hall, 2003.

Kim, Daniel H., The Link Between Individual and Organizational Learning, Sloan Management Review; Fall 1993; 35, 1; pg.37

Krackhardt, D., and Hanson, J. R., 1993. "Informal Networks: The Company Behind the Chart." *Harvard Business Review*, July-August 1993, 104-111.

Lindbeck, A. and Snower, D., 2000. "Multitask Learning and the Reorganization of Work: From Tayloristic to Holistic Organization," Journal of Labor Economics, 2000, vol. 18, no. 3

Marengo, L. and Giovanni, D., 2005. "Division of labor, organizational coordination and market mechanisms in collective problem-solving." Journal of Economic Behavior and Organization, Vol.58, 303-326.

Mark, Gloria, Gonzalez, Victor, and Harris, Justin. "No Task Left Behind? Examining the Nature of Fragmented Work" CHI 2005, Papers: Take a Number, Stand in Line (Interruptions & Attention 1) April 2-7, Portland, Oregon.

MCI Conferencing White Paper. *Meetings in America: A study of trends, costs and attitudes toward business travel, teleconferencing, and their impact on productivity* (Greenwich, CT: INFOCOMM, 1998), 3.

Repenning, Nelson P., 2000. "A dynamic model of resource allocation in multi-project research and development systems," *System Dynamics Review*; Autumn 2000; 16, 2; p. 173.

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

Wheelwright, S.C. and Clark, K. B., 1992, "Creating Project Plans to Focus Product Development," *Harvard Business Review*, 70(2), pp. 70-82.

Yaghootkar, K. and Roos, L., 2007. "The Role of Systemic Effects on Project Performance in Multiproject Environments." Whitepaper: International Conference of the System Dynamics Society.

Appendix A: Specialization Model Documentation

```
********************************
  .Control
********************************

                    Simulation Control Paramaters

FINAL TIME  = 360
        Units: Month
                    The final time for the simulation.

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

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

TIME STEP  = 0.25
        Units: Month
                    The time step for the simulation.

********************************
  .Specialization - Final
********************************

Additional Work Capacity from Overtime  =
    Table for Overtime Pressure ( Percent Loading per Person )
        Units: Dmnl
                    Additional overtime maxes out at an 80 hour workweek, or double the
normal work capacity.

Attrition  =
     Fractional Attrition Rate
        * Labor
        Units: People/Month
                    The number of people that leave per month.

Average Breadth Experience of New Hires  = 36
```

```
     Units: Month
               The amount of experience that new hires bring into the department.

Average Breadth of Experience =
   Breadth of Experience
     / Labor
     Units: Month
               Each person has the Average Breadth of Experience.

Average Depth Experience =
   Depth of Experience
     / Labor
     Units: Month
               The depth of experience per person.

Average Depth Experience of New Hires = 36
     Units: Month
               The amount of experience that new hires bring into the department.

Breadth Experience Decay Rate =
   Breadth of Experience
     * Fractional Breadth Experience Decay Rate
     Units: person
               Experience decays over time from engineers forgetting, technology
changing, or processes changing.

Breadth of Experience =
   INTEG( Increase in Breadth Experience from Hiring
       + Increase in On the Job Breadth Experience
       - Breadth Experience Decay Rate
       - Loss of Breadth Experience from Attrition ,
     ( ( Average Breadth Experience of New Hires
       * Hiring )
       + ( Labor )
         * Rate of Breadth Increase )
       / ( Fractional Attrition Rate
         + Fractional Breadth Experience Decay Rate ) )
     Units: Month*person
               Breadth of Experience is acquired through diverse project assignments in
both type and phase, the experience that new hires bring in with them and decreases due to
forgetting or technology change, or attrition.

Constant Input = 1425
```

```
          Units: task/Month
                              Input to Work Input

Depth Experience Decay Rate  =
       Depth of Experience
          * Fractional Depth Experience Decay Rate
          Units: person
                              Experience decays over time from engineers forgetting, technology
changing, or processes changing.

Depth of Experience  =
       INTEG( Increase in Depth Experience from Hiring
             + Increase in On the Job Depth Experience
             - Depth Experience Decay Rate
             - Loss of Depth Experience from Attrition ,
          ( ( Average Depth Experience of New Hires
             * Hiring )
             + ( Labor ) )
             / ( Fractional Attrition Rate
                + Fractional Depth Experience Decay Rate ) )
          Units: Month*person
                              Depth of Experience represents experience acquired for a specific job or
department.

Effect of Coordinating Cost on PDY  =
       Reference Coordination Cost
          * System Understanding Gap
          Units: Dmnl
                              The amount of loss associated with attending meetings, writing emails,
and trying to understand what others are doing and progress of the project.

Effect of Gap on Rework  =
       Table of Gap vs Rework Fraction ( System Understanding Gap )
          Units: Dmnl
                              Calculates the effect that the System Understanding Gap has on Fraction
Rework.

Effect of Switchloss on Productivity  =
       Table for Loading vs Switchloss ( Percent Loading per Person )
          * Reference Switchloss
          * Percent Scope of Work
          Units: Dmnl
```

The higher the workload per person, the greater the switching loss. The higher the scope of work, the higher the switching loss.

Exogenous New Work Generation =
Work Input
* Input Rate
+ Constant Input
Units: task/Month

Project work is an exogenous variable that is outside of the department manager's control.

Fraction Reworked =
Reference Fraction Reworked
* Effect of Gap on Rework
Units: Dmnl

The fraction of work completed that is actually done incorrectly and needs to be reworked.

Fractional Attrition Rate =
Reference Attrition Rate
Units: 1/Month

The fraction of people that leave each month.

Fractional Breadth Experience Decay Rate = 0.01
Units: 1/Month

The rate at which engineers forget, technology changes, or processes change. .

Fractional Depth Experience Decay Rate = 0.1
Units: 1/Month

The loss of job specific experience due to forgetting, technology change, or process changes.

Hiring =
1
/ 4
Units: People/Month

The number of people hired each month.

Increase in Breadth Experience from Hiring =
Hiring
* Average Breadth Experience of New Hires
Units: person

The amount that Breadth of Experience increases from new hires.

Increase in Depth Experience from Hiring =
Hiring
* Average Depth Experience of New Hires
Units: person
The rate of experience gain from new hires.

Increase in On the Job Breadth Experience =
Labor
* Rate of Breadth Increase
Units: person
The amount of experience that each person gains each month onthe job.

Increase in On the Job Depth Experience =
Labor
* Rate of Depth Increase
Units: person
The amount of experience each person gains each month.

Increase of Specialization = 0
Units: Dmnl
Degree of specialization.

Initial Breadth =
INITIAL(Average Breadth of Experience)
Units: months
The initial value when the system is in equilibrium.

Initial Labor Force = 25
Units: People
Initial people set for equilibrium.

Initial Net Productivity = 63.68
Units: Month*person

Initial Undiscovered Rework = 167.64
Units: task
Initial value set for equilibrium.

Initial Work to Do = 1676
Units: task
Initial value set for equilibrium.

```
Input Rate  = 0
        Units: task/Month
                        Input to Work Input.

Labor  =
     INTEG( Hiring
           - Attrition ,
        Initial Labor Force )
        Units: People
                        The stock of resources available.

Learning Curve Exponent  =
     LN ( 1
           + Learning Curve Strength )
        / LN ( 2)
        Units: Dimensionless
                        The amount of productivity gained for each doubling of experience.

Learning Curve Strength  = 0.5
        Units: Dimensionless

Loss of Breadth Experience from Attrition  =
     Attrition
        * Average Breadth of Experience
        Units: person
                        The amount of Breadth of Experience that people take with them when
they leave.

Loss of Depth Experience from Attrition  =
     Attrition
        * Average Depth Experience
        Units: person
                        The number of people leaving multiplied by the average depth of
experience.

Minimum Time to Complete Work  = 1
        Units: Month
                        Minimum Time it takes to complete a task while working as fast as
possible.

Net Productivity  =
     Productivity
```

```
        * ( 1
           - Effect of Switchloss on Productivity )
        * ( 1
           - Effect of Coordinating Cost on PDY )
        Units: task/(Month*person)
```

Net Productivity is net gain or loss between Productivity, Switchloss and Coordination Costs.

```
Net Work Capacity  =
     Net Productivity
        * Labor
        + Net Productivity
           * Labor
           * Additional Work Capacity from Overtime
        Units: task/Month
```

Net Work Capacity is Net Productivity multitplied by Labor plus Additional Capacity from Overtime.

```
Normalized Net Productivity  =
     Net Productivity
        / Initial Net Productivity
        Units: Dmnl
```

Net Productivity divided by Initial Net Productivity

```
Normalized Undiscovered Rework  =
     Undiscovered Rework
        / Initial Undiscovered Rework
        Units: Dmnl
```

Normalized Undiscovered Rework is Undiscovered Rework divided by Initial Undiscovered.Used to see the changes relative to initial conditions.

```
Normalized Work to Do  =
     Work to Do
        / Initial Work to Do
        Units: Dmnl
```

Normalized Work to Do is Work to Do divided by Initial Work to Do. Used to see the changes as relative to initial conditions.

```
Percent Loading per Person  =
     Work per person
        / Reference Work per Person
        Units: Dmnl
```

Fully loaded person working 40 hours per week = 100%

```
Percent Scope of Work  =
    Reference Scope of Work
      / Reorganization Trigger
      Units: Dmnl
                  Percent Scope of Work decreases as Specialization increases.

Productivity  =
    Reference Productivity
      * ( Average Depth Experience
        / Reference Experience )
        ^ Learning Curve Exponent
      Units: task/(Month*person)
                  The number of tasks that can be accomplished per amount of effort.

Pulse Quantity  = 0
      Units: Month*Dimensionless

Pulse Start Time  = 0.25
      Units: Year

Pulse Time  = 5
      Units: Month

Ramp End Time  = 10
      Units: Month

Ramp Slope  = 0
      Units: 1/Month

Ramp Start Time  = 5
      Units: Month

Rate of Breadth Increase  =
    Reference Rate of Breadth Increase
      * Percent Scope of Work
      Units: 1
                  The rate of breadth experience that each person gains per month.

Rate of Depth Increase  =
    Reference Rate of Depth Increase
      / Percent Scope of Work
      Units: 1
```

The rate of depth increase is inversly proportional with Percent Scope of Work.

Reference Attrition Rate = 0.01
Units: 1/Month
The rate at which people leave the company.

Reference Breadth of Experience =
Initial Breadth
Units: Month
The reference experience represents an all knowing person who understands the system as much as possible.

Reference Coordination Cost = 0.25
Units: Dmnl
Typically, twenty five percent of the day is spent on meetings and communications.

Reference Experience = 10
Units: Month
The experience needed to achieve Reference Productivity.

Reference Fraction Reworked = 0.05
Units: Dimensionless
Assumed that even with as much breadth of experience as possible, people are only 95% all-knowing.

Reference Productivity = 100
Units: task/(Month*person)
The productivity achievable for someone with Reference Experience.

Reference Rate of Breadth Increase = 1
Units: 1
Each person increases experience at a rate of one month per month.

Reference Rate of Depth Increase = 1
Units: 1
The depth of experience gained per month on the job.

Reference Scope of Work = 1
Units: Dimensionless
The full scope of work encompasses all tasks from design through obscolescence.

Reference Switchloss = 0.25
Units: Dmnl
A Person 100% loaded loses 25% of productivity due to switchloss.

Reference Time to Discover Rework = 2
Units: Month
The time it takes for an almost all knowing person to discover rework.

Reference Work per Person = 40
Units: task/person
The expected workload that each person can complete during a month assuming a normal 40 hours workweek.

Reorg Switch = 1
Units: Dmnl

Reorg Time = 48
Units: Month
The time at which a reorganization is triggered.

Reorganization Trigger =
1
+ STEP (Increase of Specialization ,
Reorg Time)
* Reorg Switch
Units: Dmnl
The switch to trigger reorganization.

Rework Discovery =
Undiscovered Rework
/ Time to Discover Rework
Units: task/Month
The rate at which rework is put into Work to Do is the amount of Undiscovered Rework divided by the Time to Discover Rework.

Rework Generation =
Fraction Reworked
* min (Work to Do
/ Minimum Time to Complete Work ,
Net Work Capacity)
Units: task/Month
The rate at which rework is generated.

Sine Amplitude = 0
Units: Dimensionless

Sine Period = 50
Units: Month

Step Height = 0
Units: Dimensionless

Step Time = 50
Units: Month

System Understanding Gap =
Reference Breadth of Experience
/ Average Breadth of Experience
Units: Dmnl
System Understanding Gap increases as the Average Breadth of Experience decreases.

Table for Loading vs Switchloss (
[(0,0)-(20,10)],(0,0),(0.5,0.5),(1,1),(1.25,1.15),(1.5,1.3),(2,1.5),(3,2),(4,3.9),(5,3.9),(20,3.9)
)
Units: Dmnl
Given Percent Loading per Person, generates a multiplier for Switchloss.

Table for Overtime Pressure (
[(0,0)-(20,1.5)],(0,0),(1,0),(1.25,0.1),(1.5,0.25),(2,0.4),(3,0.5),(4,0.5),(20,0.5))
Units: Dmnl
Given Percent Loaded per Person, calculates the Overtime needed.

Table of Gap vs Rework Fraction (
[(0,0)-(4,6)],(0,1),(1,1),(1.2,2),(1.5,3),(2.5,4),(3.5,5),(4,5.5))
Units: Dmnl
Given the System Understanding Gap, calculates the Rework Fraction.

Time to Discover Rework =
Reference Time to Discover Rework
* System Understanding Gap
Units: Month
Increased specialization means it will take longer to discover rework down the line.

```
Undiscovered Rework =
    INTEG( Rework Generation
         - Rework Discovery ,
       Initial Undiscovered Rework )
       Units: task
                   Stock of tasks that was done incorrectly and requires more work to fix.

Work Completion Rate =
    ( 1
       - Fraction Reworked )
       * ( min ( Work to Do
            / Minimum Time to Complete Work ,
         Net Work Capacity ) )
       Units: task/Month
                   The number of tasks that can be complete each month.

Work Done =
    INTEG( Work Completion Rate ,
       0)
       Units: task
                   Work Done accumulates at the Work Completion Rate.

Work Input =
    STEP ( Step Height ,
       Step Time )
       + ( Pulse Quantity
         / TIME STEP )
         * PULSE ( Pulse Time ,
           TIME STEP )
       + RAMP ( Ramp Slope ,
         Ramp Start Time ,
         Ramp End Time )
       + Sine Amplitude
         * SIN ( 2
              * 3.14159
              * Time
              / Sine Period )
       Units: Dimensionless
                   Generates flow of tasks into Work to Do stock.

Work per person =
    ( Work to Do )
       / Labor
```

```
          Units: task/person
                              Individual workload.

Work to Do  =
       INTEG( Exogenous New Work Generation
              + Rework Discovery
              + Rework Generation
              - Work Completion Rate ,
          Initial Work to Do )
          Units: task
                              Stock of tasks that needs to be worked on.

Work to Rework Ratio  =
       Undiscovered Rework
          / Work to Do
          Units: Dmnl
                              Work to Rework Ratio is Undiscovered Rework divided by Work to Do.
```

Appendix B: Reinventing the Wheel Model Documentation

```
********************************
 .Control
********************************

                    Simulation Control Parameters

FINAL TIME  = 50
                  Units: Month
                          The final time for the simulation.

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

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

TIME STEP  = 0.125
                  Units: Month
                          The time step for the simulation.

********************************
 .Reinvent Final
********************************

Additional Work Capacity from Overtime  =
    Table for Overtime Pressure ( Percent Loading per Person )
      * Overtime Switch
                  Units: Dmnl
                          Additional overtime maxes out at an 70 hour workweek, or 75%
the normal work capacity.

Anticipated Attrition  =
    Attrition Rate
      * Labor
                  Units: Person/Month
                          The number of people expected to leave each month based on
prior experience.
```

```
Attrition =
    Labor
      * Attrition Rate
      + Labor
        * Attrition Rate
        * Attrition Pulse Height
        * Attrition Pulse Input
                  Units: Person/Month
                        The number of people leaving the company or department per
month.

Attrition Pulse Height = 0
                  Units: Dmnl

Attrition Pulse Input =
    PULSE ( 12,
      0)
                  Units: Dmnl
                        At time=12, a pulse is sent in to increase Fractional Attrition Rate.

Attrition Rate = 0.08
                  Units: 1/Month
                        The rate that people leave per month.

Completion Rate =
    min ( Work Partially Complete
        / Minimum Completion Time ,
      Fraction Capacity to Complete
        * Work Capacity )
                  Units: Task/Month
                        The rate at which work is complete.

Effect of Switchloss on Productivity =
    Reference Switchloss
      * Table for Loading vs Switchloss ( Percent Loading per Person )
      * Switchloss Switch
                  Units: Dimensionless
                        The higher the workload per person, the greater the switching
loss.

Fraction Capacity Partially Complete =
    Weight of Work to Do
```

```
        / Total Weight
                    Units: Dmnl
                              Fraction of Capacity spent on Work to Do.

Fraction Capacity to Complete  =
    Weight of Partially Complete
        / Total Weight
                    Units: Dmnl
                              Fraction of Capacity spent on Work Partially Done.

Fraction Reinvented  = 0.3
                    Units: Dmnl
                              The fraction of work that is reinvented.

Fractional Turnover Rate  =
    Rotation Rate
        * Rotation Switch
        + Attrition Rate
        + Attrition Rate
           * Attrition Pulse Input
           * Attrition Pulse Height
                    Units: 1/Month
                              The sum of attrition and rotation rates.

Headcount Gap  =
    Max Headcount
        - Labor
                    Units: People
                              Headcount gap is the number of people allowed minus current
headcount.

Hiring  =
    Headcount Gap
        / Time to Hire
        + Anticipated Attrition
                    Units: Person/Month
                              The number of people hired per month.

Initial Input  = 150
                    Units: Task/Month
                              Reference Productivity*(Labor-Switchloss Switch*Reference
Switchloss*Initial Work to Do/Reference Work per Person) - (Reinventing Switch
```

```
                              *Fractional Reinvented*Turnover*Initial Work to
Do/Labor)*Loading

Initial Labor  = 20
                    Units: Person

Initial Loading  = 0.75
                    Units: Dimensionless
                              Loading is used to adjust initial loading per person.

Initial Work Capacity  =
     INITIAL( Labor
           * Productivity  )
                    Units: Task/Month

Initial Work Partially Done  =
     if then else ( Reinventing Switch
           = 1,
        112.5,
        Initial Work to Do )
                    Units: Task
                              Initial value set for equilibrium.

Initial Work to Do  =
     if then else ( Reinventing Switch
           = 1,
        115.19,
        Initial Input
           * Initial Loading )
                    Units: Task

Input  =
     Step Input
        + Pulse Input
           * Pulse Height
        + Initial Input
           * Initial Loading
                    Units: Task/Month
                              Input into Work Input.

Labor  =
     INTEG( Hiring
           - Attrition ,
```

Initial Labor)
Units: People
(Desired Headcount/Time to Hire+Anticipated Turnover)/(1/Time to Hire +Fractional Turnover Rate)

Max Headcount = 20
Units: Person
Max headcount is an exogenous variable set by the larger organization.

Minimum Completion Time = 1
Units: Month
Minimum time it takes to complete a task.

Minimum Partially Complete Time = 1
Units: Month
Minimum time it takes to complete a task.

Normalized Input =
Work Input
/ Initial Work Capacity
Units: Dmnl
Work input is normalized over initial work capacity.

Normalized Total Work =
Total Work
/ (Initial Work Partially Done
+ Initial Work to Do)
Units: Dmnl
Total Work is normalized over Initial Total Work in order to compare changes relative to initial conditions.

Normalized Work Partially Done =
Work Partially Complete
/ Initial Work Partially Done
Units: Dmnl
Work partially complete is normalized over initial work partially done to compare changes against initial conditions.

Normalized Work to Do =
Work to Do
/ Initial Work to Do
Units: Dmnl

```
                                        Work to do is normalized over initial work to do in order to
compare changes to equilibrium conditions.

Overtime Switch  = 1
                        Units: Dmnl

Partial Completion Rate  =
    min ( Work to Do
          / Minimum Partially Complete Time ,
       Work Capacity
          * Fraction Capacity Partially Complete )
                        Units: Task/Month
                                        Partial Completion Rate is the number of tasks started in a month.

Percent Loading per Person  =
    Work per Person
       / Reference Work per Person
       * 100
                        Units: Dmnl
                                        A fully loaded person working 40 hours per week  = 100%

Pressure to Finish  =
    1
       + Step ( Weight to Partial Trigger ,
          12)
                        Units: Dmnl
                                        Managerial pressure to complete work.

Pressure to Start  =
    1
       + Step ( Weight to Do Trigger ,
          12)
                        Units: Dmnl
                                        Managerial preference to prioritize starting Work to Do.

Productivity  =
    Reference Productivity
       * ( 1
          - Effect of Switchloss on Productivity )
                        Units: Task/(Person*Month)
                                        Prouctivity is the number of tasks that can be done given the
amount of effort.
```

```
Pulse Height  = 300
                              Units: Dmnl
                                           Multiplier for Pulse Input. 1x=300, 2x=600, 3x=900, 4x=1200,
5x=1500

Pulse Input  =
      PULSE ( 12,
         0)
                              Units: Task/Month

Reference Productivity  = 15
                              Units: Task/(Person*Month)
                                           The peak level of productivity.

Reference Switchloss  = 0.25
                              Units: Dmnl
                                           A Person 100% loaded loses 25% of productivity due to switchloss.

Reference Work per Person  = 15
                              Units: Task/Person
                                           The normal number of tasks that a person is expected to have to
be 100% loaded.

Reinventing  =
      Fractional Turnover Rate
         * Fraction Reinvented
         * Work Partially Complete
         * Reinventing Switch
                              Units: Task/Month
                                           Reinventing is the rate that work is pulled out from work partially
complete and sent back to work to do.

Reinventing Switch  = 0
                              Units: Dmnl

Rotation Rate  =
      1
         / 6
                              Units: 1/Month
                                           Rotation Rate represents the number of rotatations that occur in
a month.

Rotation Switch  = 0
```

```
                    Units: Dmnl
                            Switch to activate rotaions.

Step Height  = 0
                    Units: Task/Month

Step Input  =
    Step ( Step Height ,
      12)
                    Units: Task/Month

Switchloss Switch  = 1
                    Units: Dmnl

Table for Loading vs Switchloss  (
        [(0,0)-
(500,10)],(0,0),(50,0.25),(75,0.5),(100,1),(131.498,1.75439),(157.492,2.45614),(214.067,3.02632
),(296.636,3.42105),(374.618,3.64035),(500,3.99) )
                    Units: Dmnl

Table for Overtime Pressure  (
        [(0,0)-
(2000,1.5)],(0,0),(100,0),(125,0.1),(150,0.25),(200,0.4),(300,0.5),(400,0.75),(2000,0.75) )
                    Units: Dmnl

Time to Hire  = 6
                    Units: Month
                            The time it takes to hire someone.

Total Weight  =
    Weight of Partially Complete
      + Weight of Work to Do
                    Units: Task
                            Total weight is used to divide up capacity between Work Partially
Complete and Work to Do.

Total Work  =
    Work Partially Complete
      + Work to Do
                    Units: Task
                            Total work is the sum of Work to Do and Work Partially Complete.

Weight of Partially Complete  =
        Pressure to Finish
            * Work Partially Complete
                        Units: Task
                                Weight of Partially Complete takes into account the number of
tasks partially complete multiplied by managerial pressure to finish.

Weight of Work to Do  =
        Pressure to Start
            * Work to Do
                        Units: Task
                                The weight of work to do takes into account the number of tasks
on deck to do multiplied by management's pressure to start.

Weight to Do Trigger  = 0
                        Units: Dmnl
                                Switch for Pressure to Start.

Weight to Partial Trigger  = 1
                        Units: Dmnl
                                Switch for Pressure to Finish

Work Capacity  =
        Labor
            * Productivity
            + Labor
                * Productivity
                * Additional Work Capacity from Overtime
                        Units: Task/Month
                                Work capacity is the work that can be accomplished per month
given the number of people available and how productive they are.

Work Complete  =
        INTEG( Completion Rate ,
            0)
                        Units: Task
                                The stock of work complete.

Work Input  =
        Input
                        Units: Task/Month
                                The exogenous flow of work into Work to Do.

Work Partially Complete  =
	INTEG( Partial Completion Rate
		- Reinventing
		- Completion Rate ,
	    Initial Work Partially Done )
			Units: Task
				Work Partially Complete is an accumulation of tasks that someone
has put some effort into starting.

Work per Person  =
	Total Work
	    / Labor
			Units: Task/Person
				Work per person is the number of total tasks to do divided by
labor.

Work to Do  =
	INTEG( Reinventing
		+ Work Input
		- Partial Completion Rate ,
	    Initial Work to Do )
			Units: Task
				The stock of tasks to do.
```