Strategies for Dealing with Instabilities in a Complex, Multi-Project Product Development System Engineering Environment

by

Michael R. Wright

B.S. Mechanical Engineering, Purdue University, 1989

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
[June 2003]
February 2003

Signature of Author______________________________

Michael R. Wright
System Design and Management Program
February 2003

Certified by______________________________

Eric Rebentisch
Thesis Supervisor
Lean Aerospace Institute, Lead Researcher (Acquisition; Organizations and People)

Accepted by______________________________

Steven D. Eppinger
Co-Director, LFM/SDM
GM LFM Professor of Management Science and Engineering Systems

Accepted by______________________________

Paul A. Lagace
Co-Director, LFM/SDM
Professor of Aeronautics & Astronautics and Engineering Systems

[This page is intentionally left blank]

Strategies for Dealing with Instabilities in a Complex, Multi-Project Product Development System Engineering Environment

By

Michael R. Wright

Submitted to the System Design and Management Program
On February 2, 2003 in Partial Fulfillment of the
Requirements for the Degree of Master of Science in
Engineering and Management

Abstract

This thesis evaluates the product development process from the perspective of a multiple gas turbine engine development programs. The risk to meeting cost and schedule requirements has increased solely due to squeezing budgets and schedule to fit the "better, faster, cheaper" mold. The thesis focuses on the further risks to cost and schedule of the gas turbine product development cycle that are caused by instabilities introduced by the cyclical nature of multiple product development programs completing the cycle and new ones starting. Market and business factors influence the numbers of cycles and can not be controlled. Workload and resource-usage are not stabile within multiple product development cycles.

The analysis establishes an overview of the gas turbine engine, product development process, and project management techniques employed to deliver the product to the customer within cost and schedule constraints. The analysis then uses a risk causal framework to identify the issues that the process faces relative to the cost and schedule risk. The use of this framework identifies staffing issues to be one of the key drivers of cost and schedule risk. A systems dynamic model developed in a previous Systems Design and Management thesis was adapted to represent the product development process by adding structure and calibrating the model with realistic scenarios. The model evaluates the policies that can mitigate risks identified within the given process. Recommendations are provided within a framework that enables management to decide the appropriate use of the policies recommended

[This page is intentionally left blank]

Acknowledgements

I would like to thank my thesis advisor, Dr. Eric Rebentisch, for his excellent guidance through the thesis process. His time spent working with me through the fall semester was invaluable to the completion of this thesis.

I am extremely thankful to my supervisors and co-workers at Pratt & Whitney who allowed me the time to maximize my learning efforts though out the Systems Design and Management program. My management, Pete Chenard, Craig Bolt, and John Duffy allowed me to apply to the program and fully supported my efforts in the program.

I thank Pete Chenard, Kim Peterson, Andrea Borondy Kitts, Tom Pelland, and Jon Niemeyer, and for their input for this thesis.

I would also like to thank my co-workers Ted Mattern, Steve Rose, Jen Voellinger, Nick Hoffman, Jon Campbell, and Greg Gower who took on added responsibilities during my school efforts.

Above all, I thank my wife Kris who was highly supportive through the entire program while starting a new career. Her help during my major knee injury while writing this thesis allowed me to complete the thesis with minimal impact.

[This page is intentionally left blank]

Table of Contents

List of Figures

List of Tables

1 Introduction

1.1 Problem Statement

Data obtained from past gas turbine product development programs indicate that on average new gas turbine engine program development costs typically run significantly higher than original budget allocations. Recent historical time frames for a gas turbine engine product development process have typically been 48 months. Currently planned product development process time frames are being squeezed down to as low as 18 months. The reason for the reduced schedule relative to achieved averages in cost and schedule is that during the last decade aerospace companies have faced a decline in government spending and a reduction in available resources for product development programs. The aerospace industry reaction to the decline has been a drive toward "better, faster, cheaper" as coined by NASA's recent leader Dan Goldin. Lean practices and quality initiatives, such as Six Sigma; have allowed companies to make great strides in improvements in efficiency and productivity levels while also increasing quality. Lean Initiatives are being used identify to waste and eliminate the bottlenecks. But since "Better, Faster, Cheaper" emerged during the mid 1990's, management teams have equated uses of these tools as methods to enhance productivity and therefore reasons that more work can be accomplished with less people. Even with the use of the management tools mentioned above instabilities are still present in the product development process. Annual budget cycles, requirements creep, unrealistic program cost estimates, and technology readiness are a few problems that beset program development programs and cause instability. As the aerospace companies become more efficient and utilization of resources is maximized, the consequences of program instabilities become more profound. Pratt & Whitney is currently facing the effects of the use of lean practices to create a lean enterprise in conjunction with instabilities inherent in the gas turbine engine product development cycle. Two major development programs finished their respective product development cycles in mid to late 2001. The ends of the cycles were characterized by a high degree of resource utilization in the validation phase of the product development process. Recently, three new programs initiated the concept development phase of the Product Development Process. The inherent instability experienced by the Systems Engineering - Validation group caused by high resource utilization in the validation phase, transitioning to relatively lower levels in concept and preliminary design phases, and back to high utilization as the product development process again transitions to the validation phase is the main problem studied within this thesis.

1.2 Motivation

Currently Pratt & Whitney faces a major challenge to their product development processes. Three major product development programs have been undertaken potentially causing instability in the workload and corresponding resource requirements at the company. The promises to the customer to provide new products on schedule in conjunction with a low development cost given the presence of instability presents a great challenge to the product development system. The time that has been allotted to the product development process so that the product can be delivered to the customer when promised are aggressive compared to the achieved past averages. The drive to minimize costs within the company cause resources to be matched with workload. Staffing is reduced when workload is low before the three projects move into resource intensive phases of the process.

The thesis will outline that the result of undertaking the three projects following reductions in staff may cause the inability of the system to achieve the cost and schedule requirements. Relative to the company's product, organization, and processes there is little insight into the possible causal factors of cost and schedule growth. The thesis will evaluate the system relative to risk drivers to evaluate the nature of the problem. Beyond identification of the causal factors the thesis will evaluate the effect of the problems to the ability of the projects to deliver on promises to the customer and the ability of remediation efforts to effect the system. The thesis will enhance and calibrate a system dynamic model of multi-project product development programs to the planned efforts at the company. The model will be used to assess how bad the problems can become as the product development processes are beset with lower productivity and quality than planned. In addition the effect of the normal introduction of major late problems will be evaluated. Finally the thesis will assess the effectiveness of various remedies in overcoming the problems inherent in the process. The following sections outline the product development processes that have been undertaken at Pratt & Whitney. The processes are rolled up into an overall picture of the workload profile over the life of the programs. The situation described below will be used as the baseline calibration of the plan for the system dynamic model.

1.2.1 Project X

Project X is a major defense gas turbine engine development program for a newly designed aircraft awarded in 2001. The project was awarded in late 2001 and the first flight ready propulsion system is slated for delivery in late 2005. The preliminary design phase was completed in mid 2002 while detail design is slated to be complete in early 2003. The first validation engine will be delivered to test in late 2003. The head of the aircraft system program believes the product development process is under a "very aggressive schedule".[1] The propulsion system uses the high spool of an existing design. The rest of the propulsion system is newly designed. The complexity is the propulsion system has been highly integrated into the flight control system and now represents a much more integrated piece of the aircraft system than ever before.

1.2.2 Project Y

Project Y is a commercial engine program for a newly designed aircraft that is an alliance between Pratt & Whitney and another major gas turbine engine manufacturer. The integration of Pratt & Whitney's low spool and the other company's high spool provides the basis for the design. Overall

---

[1] Wall, Robert and Fulghum, David A., Lockheed Martin Strikes Out Boeing, Aviation Week and Space Technology, Oct. 29, 2001

engine systems such as the control systems, fuel and oil systems, and cooling systems must also integrated during the product development process. The concept definition and optimization phases of the program were complete in mid 2002. Preliminary design was complete at the end of 2002. The detail design phase will continue through most of 2002. The first validation engine will be delivered to test in early 2004. The first flight ready propulsion system will be delivered to the customer in the third quarter of 2005 that will result in a first flight of the aircraft system in late 2005. The new engine will "bring significantly lower operating costs" and will "be able to provide substantially better weight, fuel burn, noise, and cash operating costs" [2]

1.2.3 Project Z

Project Z is also a commercial engine program. The engine was a newly designed, full engine, product development program. The engine was initially designed and tested in the late 90's and early 00's. In 2002 the engine initiated flight testing on a newly designed aircraft. The engine fell short of performance expectations and could not be sold. The new program introduces a new compressor design and a great deal of other improvements to meet promised performance goals. The program completed preliminary and detail design during the rest of 2002 and the first validation engine to test will be delivered in early 2003. The first flight ready propulsion system will be delivered to the customer in late 2004 and flight testing of the aircraft/engine system will begin

All three of these programs have very similar scheduled delivery promises to the customers. The programs promised completion dates are faster than the historical development times the company has demonstrated during the past. The company has bet that process improvement systems will enable product development programs to be done "Better-Faster-Cheaper" by improving productivity and quality within the product development process. The programs have been scheduled with very

[2] Engine Alliance Press Release, GE and Pratt & Whitney team on new engine for Growth 747, Evendale, OH, May 8,

similar timelines and as a result the resource utilization profiles stack up to create a high demand for resources. The peak demand follows a relatively low demand for resources during which the company moved to reduce staffing to better match resources with workload. The situation may lead to instability in the resource curve. The situation is graphically represented in Figure 1. In addition to instability presented by the resource utilization the programs are also dealing with normal program instabilities. The motivation for this thesis is to help identify potential risks to the programs that are not being addressed by the current continuous improvement programs and to identify remedies to these challenges.

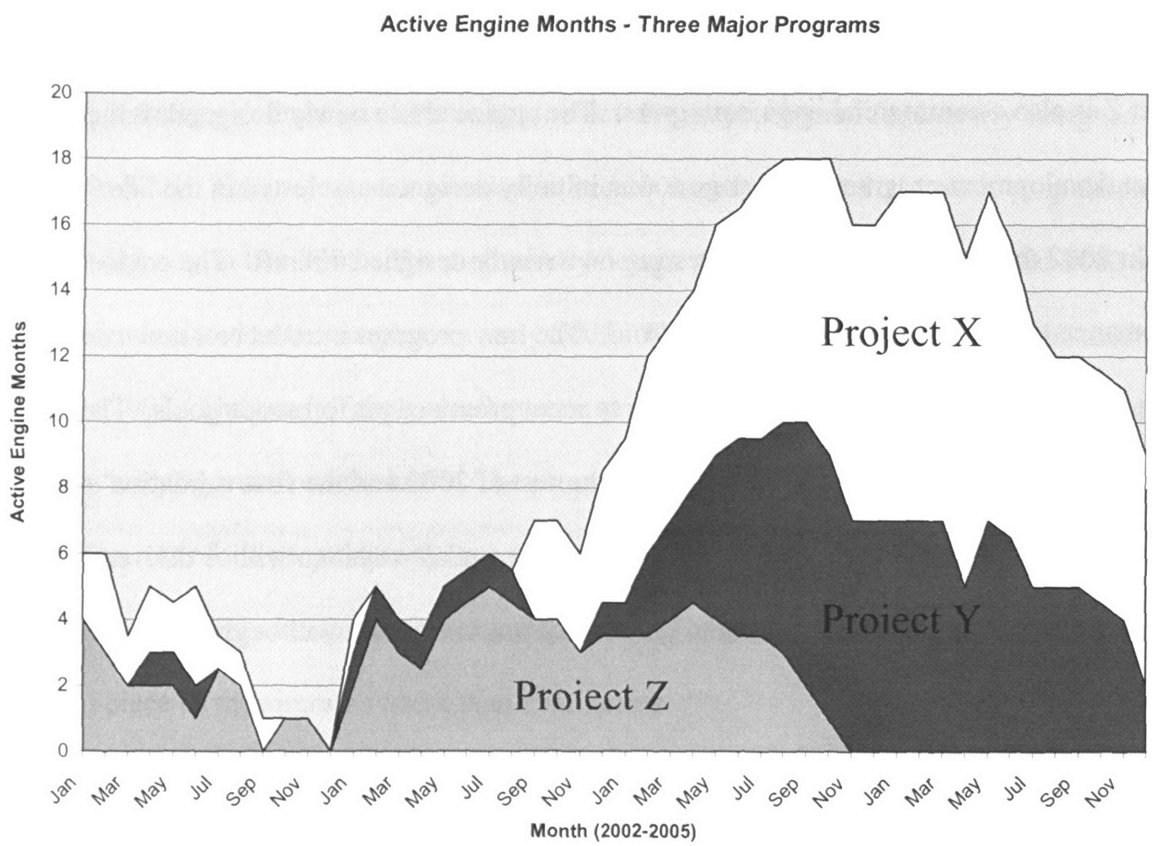

Figure 1-1 : Work To Do for the Three Programs

1996

1.3 Scope/Goal

This thesis is concerned with the product development process of a highly complex system. The process is currently being squeezed to drive higher system performance and deliver value to the customer. Based on a paper prepared for the Lean Aerospace Initiative in September 2000 Value can be defined as

$$Value = \frac{f_P(performance)}{f_C(\cos t) \bullet f_t(time)}$$

This can be interpreted such that increased performance (Better), Shorter Time (Faster), and Lower Cost (Cheaper) lead to increased value.[3] Therefore the system performance parameters are performance, cost, and schedule. For the product development process performance is the ability to meet all customer requirements. Cost translates to the amount of money that is spent developing the final product. Schedule is a measure of the time required to execute the given program to deliver a flight-qualified engine to the customer for flight-testing.

The end customer for Pratt & Whitney as a company would be the U.S government for military product and the airlines for the commercial product. The military customer contracts with the internal program office for delivery of the defined product in the given time frame at the given cost. The commercial customer usually contracts with the airframe manufacturer to provide a new centerline or derivative powerplant for initial sales. Engines are actually sold to specific airlines, but the customer requirements are defined by the aircraft system. The thesis will also show that the Integrated Program Management Office is tasked for delivery on cost and schedule metrics on any given program. For the purposes of this thesis it is assumed the goal is to deliver to the customer the required performance and that goal will be achieved regardless of cost or schedule. As in any mature

[3] Murmann, Earll M., Walton, Miles and Rebentisch, Eric. CHALLENGES IN THE BETTER, FASTER, CHEAPER ERA OF AERONAUTICAL DESIGN, ENGINEERING AND MANUFACTURING, Lean Aerospace Initiative Report Series RP00-02 (September 2000)

industry incremental change is the driver for enhanced performance in new products. Performance improvements over the years in commercial gas turbine engines can be seen in Figure 1-2.

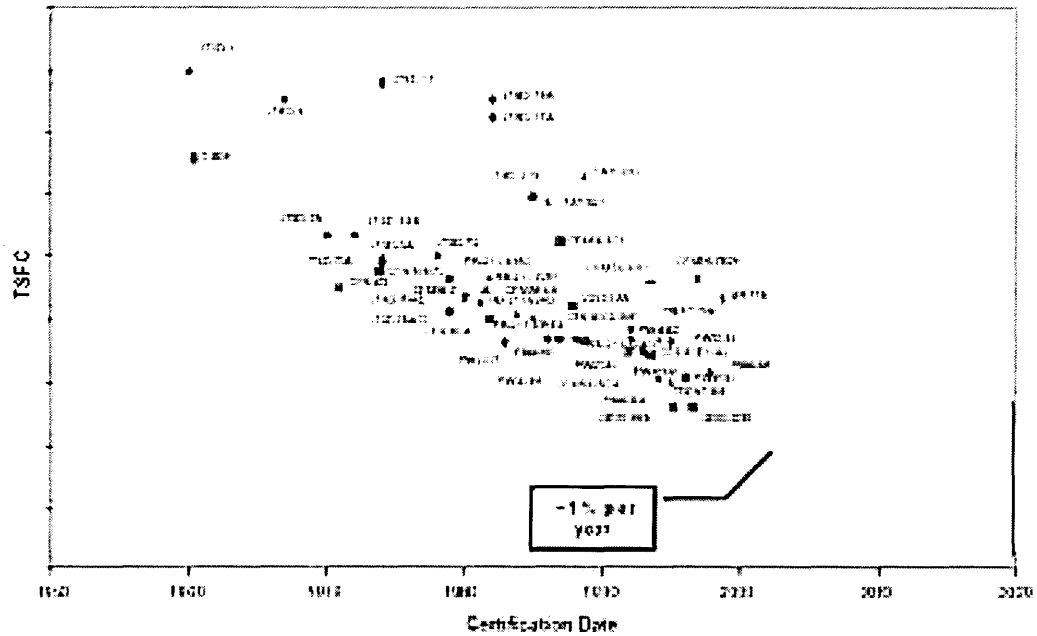

Figure 1-2: Rate of TSFC improvement over time for the three major engine manufacturers: Pratt & Whitney, General Electric, and Rolls Royce

In the case of the new programs outlined in this thesis, the performance goal of all three programs is higher than that of existing engines and therefore is inherently better.

The next paragraphs will describe the body of the thesis that will follow. Section 2 of the thesis will describe the research method used to evaluate the product architecture, organization, and processes for the key drivers in the risk to delivery of value to the customer. An overview of the literature reviewed for the creation of the thesis will be presented. A framework for the identification of risk in a complex product development process will also be introduced in Section 2.

Section 3 has the primary goal of providing the background that will define the company organization, product architecture, and identified processes. In this section a systems engineering approach of identification of the system of systems that comprise the overall system is used to evaluate the ability to develop the products. The definition provides the baseline system that will be evaluated using the risk framework.

Section 4 presents the risk framework as defined by Tyson Browning. The framework takes the defined system and evaluates the various causal relationships of variations in uncertainty in completing complex product development. The major goal is to relate the risks to cost and schedule performance and identify the issues present within the system that can be varied to mitigate risk to cost and schedule performance. The section also defines why cost and schedule are the two key metrics within the gas turbine product development process rather than the performance, technology, market, and business risk. With the delivery requirements pushed to be faster than previously demonstrated, one significant failure during the process can have catastrophic effects to cost and schedule. We have noted this occurrence in the project Z above. Risk and value are intimately interrelated, as the quality of the value metric is related to the probability of certainty of its representation.[4] The section demonstrates that resources or staffing are a key driver in the system dynamics within the product development process at Pratt & Whitney. Issues related to staffing such as the use of overtime, outsource resources, and the amount of available resources are identified as key drivers. Also, due to the use of quality initiatives and process improvement tools, productivity and quality are revealed as other key drivers that must be evaluated by the system dynamic model.
Section 5 of the thesis introduces the key causal loops that were used by Karl Pilon and Greg Herweg to develop a multi-project, product development, system dynamic model to evaluate the system dynamic effects of staffing decisions. The underlying assumptions that the model is based on are related to the defined system at Pratt & Whitney. The model's applicability to the Pratt & Whitney System is defined.

Section 6 defines the changes required to the system dynamic model that calibrate the model to the planned performance of the system and defines the baseline. The system is then evaluated using real system performance metrics of productivity and quality that are less that what is planned. In addition

[4] Murmann, Earll M., Walton, Miles and Rebentisch, Eric. CHALLENGES IN THE BETTER, FASTER, CHEAPER ERA OF AERONAUTICAL DESIGN, ENGINEERING AND MANUFACTURING, Lean Aerospace Initiative Report

the real baseline performance of the system is set when the projects are beset by problems late in the process which create an added 3 months to the process once all of the dynamics are in effect. The three-month delay is an average delay for a late redesign in the product development process. The policies related to utilization of outsource labor, improvements in productivity and quality, and planned resource utilization at the beginning of projects are then evaluated.

2 Research Method

The paper will attempt to identify policies for enhancing the probability for success given the current program by

1) Literature review - Consultation of literature in attempt to better frame and define the issues that are inherent in the current product development process.
2) Definition of the organization, processes, and product architecture by using personal experience, interviews, and data review to relate the Pratt & Whiney product development process to the overall risk in system performance.
3) Based upon the system of systems defined in the above step the thesis will identify various policies carry out sensitivity testing using a modified version of the system dynamics model for multi-project product development program developed by Greg Herweg and Karl Pilon.
4) Evaluation of the system performance using the results gained by using the model will be performed relating to the ability of the various policies to reduce cost and schedule risk..
5)

2.1 Literature Review/Related Work

A review of literature was performed to provide better insight into the issues facing Pratt & Whitney. Surrounding research was performed on Program Instability, Firefighting in a multi-project enterprise, Risk and Uncertainty in Complex Product Development, Process Management, and Manpower Staffing decisions in Multi-Project enterprises. Initial research centered on Instability and the impact on complex technical systems. Research by Cutcher-Gershenfeld and Rebentisch[5] centered on moving from traditional buffering strategies toward more flexible "lean" practices for

[5] Cutcher-Gershenfeld, Joel and Rebentisch, Eric, "The impact of Instability on Complex Social and Technical Systems", MIT Engineering Systems Division Internal Symposium (April 2002)

dealing with instability. Previous research by Rebentisch[6] centered on factors driving program instabilities. The research identified budget or production rate changes, requirements changes, and unanticipated technical challenges as top drivers of program instabilities based on surveys completed by aerospace systems development centers associated with the Air Force, Army, and Navy. The research brought up many of the issues seen at Pratt & Whitney. In particular, the main driver being budget and production rate changes could be seen as representative as the instability represented by the situation at the company. The research listed top mitigation strategies as use of Integrated Product Teams, management reserve, co-location, and use of computer aided tools for scheduling, modeling, and design. All of the mitigation strategies are in use at the company. The research suggested that the best performing strategies placed emphasis on flexibility and risk management as well as and a shared view of the system and the ways to respond to instability. Since the mitigation strategies seemed to be already in use a means to evaluate the system at Pratt & Whitney for weaknesses was desired to be found.

The author turned to research by Tyson Browning surrounding Modeling and Analyzing Cost, Schedule, and Performance in Complex System Product Development[7]. The Ph.D. thesis presented a causal framework with which to analyze the current state of the on-going efforts for dealing with program instabilities at Pratt & Whitney. The framework will be elaborated on in later sections to evaluate the company's product development system. The framework was utilized in this thesis because the framework provides the most comprehensive look at the product development process relative to risk when highly complex products are being developed. The framework provides a filter with which to evaluate the process and garner the key drivers that can be effected to mitigate cost and schedule risk with in the product development process at Pratt & Whitney.

---

[6] Rebentisch, Eric, "Preliminary Observations on Program Instability", Lean Aircraft Initiative White Paper-Lean 96-03 (October 1996)

[7] Browning, Tyson R., "Modeling and Analyzing Cost, Schedule, and Performance in Complex System Product Development", Doctoral Thesis, (1998)

The thesis research also looked closely at the issue of Fire Fighting in new product development brought up by Repenning[8] and followed by work by Repenning, Goncalves, and Black[9]. The research centers on how organizations descend into fire fighting and how given circumstances drive the organization into a downward spiral. The authors believe that the causes of fire fighting are the reallocation of scarce resources to programs that are further in the product development process. The program that gave up resources will fail to complete tasks and a vicious cycle begins where resources are only allocated for high priority tasks at the detriment to other projects. The only way to relieve the fire fighting cycle is to provide adequate resources. The instabilities at Pratt & Whitney result in a shortage of resources. High overtime use and low morale caused by a fatigued workforce are evidence of the shortage of resources. Employees describe work/life split as work/work split. When problems arise with a product the company will aggressively attack the problem with all required resources. As we will see, much of the current efforts at the company are directed at becoming better at planning and more up front knowledge of required tasks to be accomplished. Based on the above research and the given situation at Pratt & Whitney adequate staffing could be seen as a major of risk and would need to be further evaluated. Potential policies to reduce cost and schedule risk would involve staffing.

Other research centered on Process Management techniques and their relativity to the current situation. The literature review looked at three papers on Process management techniques. Research by Stefan Thomke looked at rethinking R&D in terms of the way complex experiments are conducted and the organizations required that support rapid experimentation[10]. The literature outlined methods to enable faster iterations in the development process. One method defined was utilization of new

---

[8] Repenning, Nelson P., "Understanding Firefighting in New Product Development", Journal of Product Innovation Management, (March 2001)

[9] Repenning, Nelson P., "Past the Tipping Point: The Persistence of Firefighting in Product Development", Sloan School of Management, (2002)

[10] Thomke, Stefan, "Enlightened Experimentation, The New Imperative for Innovation", Harvard Business Review (Feburary 2001).

technologies to conduct complex experiments quickly and cheaply. This method reiterated the findings in the survey of aerospace development programs. Front loading of development testing provided information on unanticipated iterations that could be solved at less cost and schedule risk. Organization for rapid experimentation involved use of cross-functional teams. All of these methods have been used at Pratt & Whitney to help the product development process. One caution brought up in the paper is the issue of resource utilization. Resource utilization levels effect the ability of the development process to respond to rapid increases in activity levels without creating bottlenecks. Utilization levels will be analyzed using the systems dynamic model to determine the ability of the organization to handle the instabilities present in the defined system. Two papers by Adler, Mandelbaum, Nguyen, and Schwerer delved in to the subject of the product development process as a stochastic processing network in which engineering resources are workstations and projects are jobs that flow between the workstations[11]. The second paper focuses on lessons learned from lean manufacturing and applied to new product development[12]. The papers focus on evaluating the product development process from a perspective of efficiency. The development process needs to have data collected on the time required to complete tasks involved in the development process so that methods of improving the effectiveness of the process can be found. Process management techniques depend on evaluation of tasks and task length. Productivity can be enhances. The effect of productivity enhancements and lower utilization of resources for bottleneck alleviation are policies that were derived from the research presented in the papers.

---

[11] Adler, Paul S., Mandelbaum, Avi, Nguyen, Vien, and Schwerer, Elizabeth, "From Project to Process Management: An Empiracally based Framework for Analyzing Product Development Time", Management Science, Vol 41, No 3 (March 1995

[12] Adler, Paul S., Mandelbaum, Avi, Nguyen, Viem, and Schwerer, Elizabeth, "Getting the Most out of Your Product Development Process", Harvard Business Review, (March April 1996)

The literature review then evaluated research done in systems dynamics modeling by Greg Herweg and Karl Pilon[13]. The thesis explores different policies and their effect on manpower and project completion dates. Most of the policies explored by Herweg and Pilon could not be applied in Pratt & Whitney's current system. The risk framework outlined by Tyson Browning identified that the business and market forces are key drivers of uncertainty in the product development process. These forces do not allow the main conclusions of the Herweg-Pilon thesis to be applied. Deferring projects to smooth out the workload or eliminating the amount of projects are impossible for the company to execute and continue to be a viable player in the gas turbine market. But the model could be used as the basis to explore other policies regarding staffing that could be applied because the base causal loops that drive staffing in the model are directly applicable to the Pratt & Whitney system.

2.2 Framework

The product development process for complex systems presents many challenges to produce value to the customer. The current situation at Pratt & Whitney has management driving to control this process and deliver value to the customer. The value proposition of the gas turbine product development program is better performance at lower cost within a faster schedule. Inherently, the complexity of the process and the commitments made to the customer relative to performance, cost, and schedule drives a high level of risk to the company. The development of a new centerline commercial gas turbine engine has frequently been equated to "betting the company" due to the high costs of development and the highly competitive selling atmosphere that causes product price to be set to a minimum. The lifecycle of the gas turbine engine has historically been around 30 years. The long lifecycle drives the time to recoup the costs of the development program beyond the normal accounting vision.

[13] Herweg, Gregory M. and Pilon, Karl E., "System Dynamics Modeling for the exploration of Manpower Project Staffing Decisions in the Context of a Multi-Project Enterprise", System Design and Management Masters Thesis, (February 2001)

Because of the factors mentioned, the product development process for the gas turbine engine becomes an exercise in managing risk. Risk as defined by Browning is the uncertainty regarding the product performance in the marketplace and the ability of the development process to deliver that product within the given schedule and budget.[14] Literature review revealed a framework for defining a program in terms of the principle sources of risk based on the causal relationships between categories of risk and the key drivers. Since Pratt & Whitney's current management practices are focusing on using quality initiatives to drive program cost, schedule, and quality toward the given goals, the research by Tyson Browning that defined a framework for evaluating program risk was deemed to be the best method to identify weaknesses within the system. Risk Reduction at Pratt & Whitney is directly related to the categories outlined in the literature. The framework looks at performance, development cost, and schedule uncertainties. The use of this framework will guide and focus the analysis in the areas determined to have weak risk mitigation plans.

[14] Browning, Tyson R., "Modeling and Analyzing Cost, Schedule, and Performance in Complex System Product Development", Doctoral Thesis, (1998)

3 Background

Background is presented here to define the Gas Turbine Product Development Process. The definition level of complexity that exists in the process speaks to the amount of risk the program undertakes to meet cost, schedule, and quality goals. Systems engineering principles demand that the systems of systems be evaluated when analyzing a system for problem mitigation. According to Boppe the system of systems approach requires evaluation of three linked systems.[15] The definitions of the three systems are Product (Architecture), Process, and Organization. The background section of the thesis will define these linked systems to define the basis with which to evaluate system performance. The definition of the system presented here will be used to relate to the risk framework that follows. The complexity of the architecture drives an organizational structure that enhances the ability to mitigate risk. The processes are well defined and thus serve to reduce risk levels. So, the reader must know the definition of the system so that the references to the ability of the company to mitigate risk can be drawn.

3.1 Gas Turbine Engine System Architecture

To provide a framework to understand the complexity of the gas turbine engine and the resulting work to do structure one must first understand the basics of gas turbine engines. The complexity of the engine drives the organization and the need to have well defined processes. It is the modularity that drives the ability to utilize Integrated Product Teams at the component levels and below. The integrality if the engine drives the need for strong systems organizations. Without an organizational structure that is defined with the above components the risk to cost and schedule will be greatly increased. The analysis utilizing the risk framework will define product complexity and degree of activity coupling as major drivers of cost and schedule risk. A gas turbine engine is a turbine engine

[15] Boppe, Charlie, "ESD.33J Systems Engineering Class Notes", MIT, (June 2001)

that runs on gas rather than steam or water. The gas that operates the turbine is the product of combustion that takes place when a fuel is mixed and burned with air passing through the engine. The gas turbine engine uses the theory of jet propulsion to carry an air vehicle aloft. Jet propulsion is the propelling force generated in the direction opposite to the flow of a mass of gas or liquid under pressure, which is escaping through an opening, called a jet nozzle. The propelling force is known as thrust. The purpose of an aircraft gas turbine engine is to generate a propulsive force greater than the drag forces associated with the aircraft and propulsion system combination. There are three types of gas turbine engines. The difference between them is the mechanism by which thrust is derived.

- **Turbo-Jet** engines derive their thrust from the jet of exhaust exiting the engine core at the rear of the engine;
- **Turbo-Fan** engines derive their thrust from a combination of by-pass flow (i.e. the portion of flow that does not pass through the engine core) and exhaust jet;
- **High By-Pass Turbo-Fan** engines having large diameter Fans that provide the majority of the overall thrust produced by the system. Most of the incoming flow does not pass through the core of the engine; it instead bypasses the engine core to directly produce thrust.

There are differences in the decomposition between military and commercial engines. In general military engines are Low By-Pass Turbo Fans and Commercial engines are High By-Pass Turbo Fans. The reason for this is the need for high efficiency for the commercial application and the need for fast transient response in the military application.

In the context of the aerospace industry a gas turbine engine provides the functions of producing thrust at low weight and low cost to the customer. Cost can be further split into acquisition, operational, and maintenance cost. Thrust Specific Fuel Consumption is the measure of operational cost. Reliability is the measure of maintenance cost. The system provides value to the customer by maximizing thrust, minimizing weight, and minimizing cost for the specific application. Whether the customer is Military or Commercial the functions must be traded to find the optimal solution for the application. Much of this type work is done in the concept development phase of the product development cycle. For example, in the case of the military customer high levels of thrust are delivered by addition of an augmentation system. The augmentor can provide high levels of thrust by penalizing TSFC. When in military terms speed is life, this is seen as an acceptable trade to add such a system to the gas turbine engine. Although in recent years, the ability to push a military aircraft past mach 1 without the use of an afterburner (i.e. Supercruise) has been a requirement to allow greater range and time over target. In the case of commercial engines an engine may be made more maintainable at a slight cost to TSFC. Depending on the relative cost of fuel, this trade may be seen as valuable.

The systems architecture decomposition of the form of the gas turbine engine is fairly standard and is the basis for what defines the most efficient means of workflow. The following section describes the modularity and integrality of the gas turbine engine.

3.1.1 Modularity and Integrality

To understand the how the system of systems are interrelated at Pratt & Whitney it is useful to understand the common system decomposition of the gas turbine engine. Fundamentally, there are three main functions, which enable a gas turbine engine to produce propulsive thrust: compression, combustion, and expansion. These functions are normally decomposed further to provide the most modularity between the components. The first level decomposition is standard to the industry. Multiple theses by Mascoli, Moy, Bartkowski, Rowles, and Hague have used this decomposition as the basis for Design Structure Matrices (DSM) studies. The DSM studies analyzed the degree of modularity of the gas turbine engine architecture and the degree of interactions that cause coupling within the design. The way in which the gas turbine engine is decomposed to create a more modular system helps define the most efficient means of workflow.

Based on industry standards and the above work the typical system architecture decomposition is as follows.

Compression Systems

Fan* - A low compressor (Fan) which compresses the air to a higher-pressure level. The fan exit air is divided into two portions. The outer air is called bypass air and goes directly to an exit nozzle and this high velocity air produces thrust. The inner compressed air is called core flow and enters the core of the engine or the Low Pressure Compressor.

Low Pressure Compressor (LPC)* - First of two compressors which compresses air for entry into the core of the engine. Dual spool compressors result in very high compressor efficiency, compression ratio, and thrust.

* - For Military applications the Fan and LPC are combined into one module.

High Pressure Compressor (HPC) - In the core of the engine the high compressor further compresses the core airflow to a relatively high-pressure ratio. The air temperature is raised. For commercial applications only a small percentage of airflow is directed through the HPC.

Combustion System

Diffuser and Combustor - The air then enters a combustor, which mixes the high-pressure and temperature air with fuel and combusts the mixture. This produces an extremely high temperature; high-energy gas that enters the turbines and is further used for propulsion of the aircraft

Expansion System - The turbines then diffuse the air and convert the energy into mechanical work. A shaft connects the turbines to the compressors and the mechanical work is used to power the compressors.

High Pressure Turbine - The high pressure turbine drives the high pressure compressor and must survive in the harshest conditions in the engine.

Low Pressure Turbine - The low pressure turbine drives the low pressure compressor and fan modules.

Exhaust System - After passing through the turbines, the hot air is forced through the exhaust opening at the back of the engine. The narrowing walls of the exhaust force the air to accelerate. The weight of the air combined with its acceleration drives the engine and the airplane attached to it, forward. The military application has the option to add fuel and mix the fan bypass air back into the exhaust stream to initiate augmentation and increase the acceleration for extremely high amounts of propulsive force.

Figure 3-1shows the cutaway of a gas turbine engine indicating the modules described above.

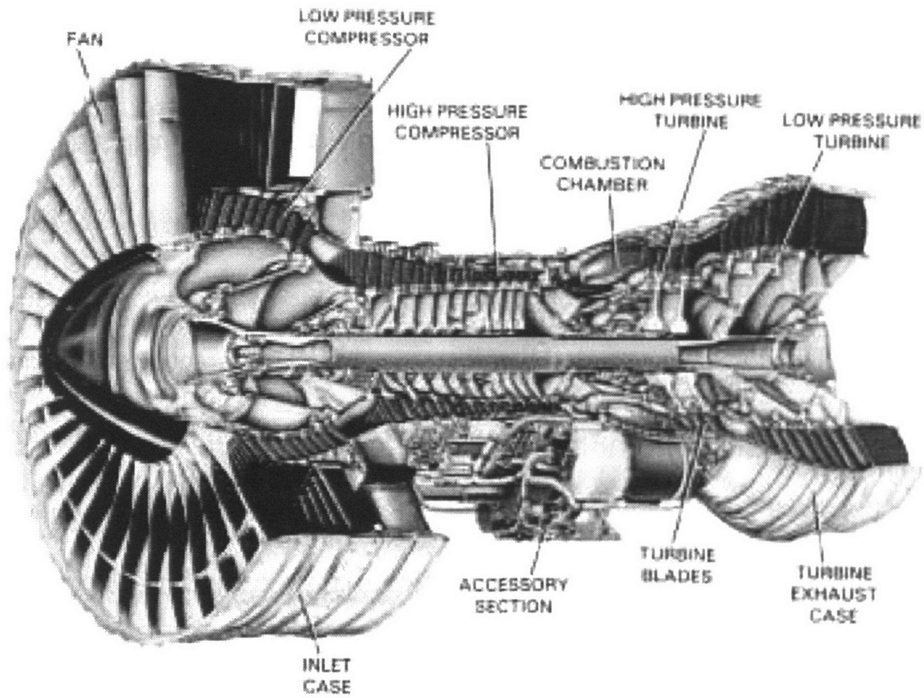

Figure 3-1 Cutaway of a commercial gas turbine engine

In addition to the modules described above the following modules are included the component breakdown and organizational structure of the engine development process.

Electrical and Mechanical Systems

Mechanical Components - Mechanical systems such as bearings, seals, shafts, and gearboxes that allow the aerodynamic components to operate.

Externals and Control System - The externals distribute fuel, oil, hydraulic pressure, and electrical signals to the required components of the engine. The control system includes the Full Authority

Digital Electronic Control (FADEC), the control actuators, and sensors. The control system controls fuel distribution, variable geometry, and cooling systems on the engine.

The DSM's by Mascoli show the above decomposition of the gas turbine engine demonstrate both integral and coupled system responses. The modular decomposition minimizes the coupling between the subsystems and is the reason that the companies have evolved to organize around the defined composition. The DSM also demonstrates the gas turbine product development process is highly complex due to the volume of the design parameters that must be satisfied. The complexity of the process requires efficient and timely information flow to effectively manage the system. Because of the nature of the information flow requirements, this workflow has driven the organizational structure evolution at Pratt & Whitney over the years. In contrast to the integral nature of the modular decomposition, the DSM studies revealed literally thousands of interactions that could affect system level performance. Two of the main value propositions for the customer, performance and product cost are highly dependent on the interaction of all of the subsystems. It is this coupling which leads to the difficulty and complexity of the product development process. The amount of work required developing a new centerline gas turbine engine and the difficulty in finding problems early in the development process drive the amount of rework that becomes necessary during the process. The thesis by Bartkowski elaborates that there may be many possible solutions to achieving the customer value and none can be identified as unique or optimum.[16] The process that arrives at the solution is iterative due to the fact that the process involves returning to earlier steps when concepts or designs are found to be flawed. Moscoli believed that due to the modular decomposition of the engine the component centers were able to work in relative isolation once the requirements were defined. Once the components were developed and produced they could be integrated into the system and evaluated. Problems at this point in the process greatly increased the time required to rework the designs. These rework cycles are unintentional iterations in the process. The coupling in the process gives rise to the requirement of systems organizations that will enhance communication between the component organizations and identify iterations earlier in the process. We will see in section 3.3 that Pratt & Whitney has formed the organization around the modularity of the gas turbine engine and utilized a systems organization to deal with the coupling that is inherent between the modules. The organizational changes work to reduce the uncertainties inherent in developing a modular product that exhibits a high degree of coupling at the system level.

3.2 Product Development Process at Pratt & Whitney

The product development process is defined as the sequence of steps or activities, which an enterprise employs to conceive, design, and commercialize a product (Ulrich and Eppinger, Product Design and Development, SE). The purpose of this section is to demonstrate that the processes are highly documented and deliverables are defined. The risk framework will show that design specifications, requirements quality, simplicity, and stability as well as communication, coordination, and integration in the process are key drivers in the process. Well-documented processes can enhance the ability to deliver quality requirements that are stable. The communication will be enhanced if the processes are known and deliverables are defined. Definition of the process allows better definition of the tasks, activity set completeness, and activity sequence quality so process length and sequencing can be determined with less uncertainty. In addition the schedule and cost risk can be raised without a robust review process. The definition of the process reveals the high degree of reviews within the process.

With the advent of ISO 9000 practices for quality assurance all processes are documented to obtain and keep certification. System Level Procedures were defined to document the Product Development process. The procedures were first implemented during mid 90's and have been undergoing

---

[16] Bartkowski, Glenn, "Accounting for System Level Interactions in Knowledge Management Initiatives", SDM Masters, (February 2001)

refinement through the years. At Pratt & Whitney one documented process governs the Product Development Cycle for all Commercial, Military, and Industrial Products. The System Level Procedure is known as the Integrated Program Deployment (IPD) Process. The IPD process is the overarching business process for all levels of systems architecture decompositions (Part, Component, or System). Through IPD, customer and internal business requirements are integrated across the business functions for the entire program life cycle from conceptual design to the end of service. The IPD process divides the life of a program into 6 key activities. The steps that define the program phases are:

Concept Initiation

Concept Optimization

Preliminary Design

Detailed Design

Validation

Delivery, Service, Support

To deliver a quality product management utilizes program and product reviews that are tied to the phases and provide a forum for measuring progress toward stated program objectives. Program reviews are a formal executive-level management gated review process applied to engine programs and other business ventures that verifies accomplishment of performance, market share, ability to be manufactured, serviceability, and other requirements against the business case, technical objectives, and customer requirements. Product reviews are a system, module or part review conducted to evaluate an engine design, its development program and operational status to requirements. The alignment of these reviews with the IPD process is shown in the figure below.

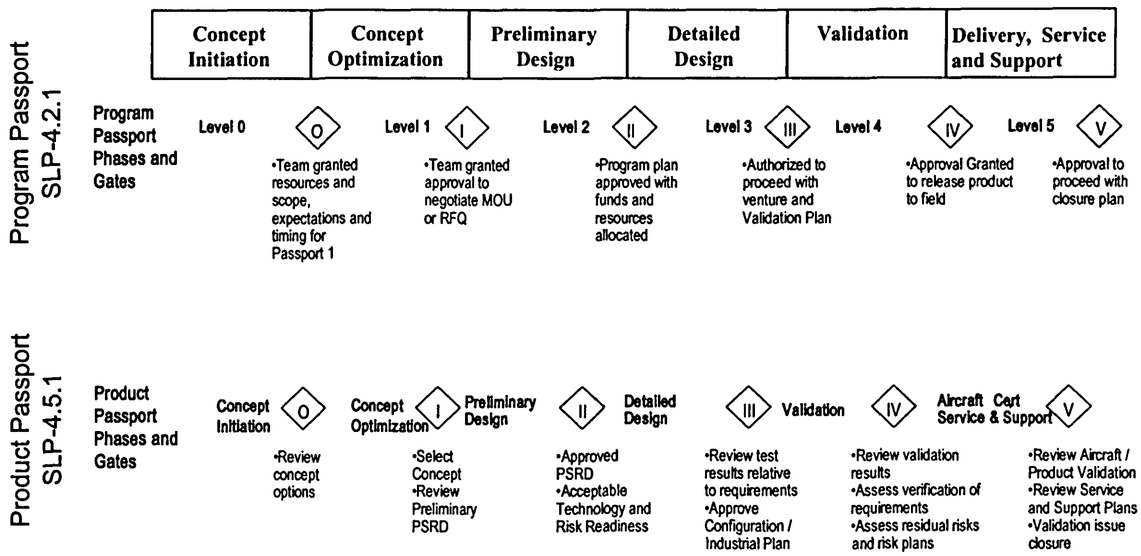

Figure 3-2 : IPD Processes, Phases, and Supporting Passport Reviews

For the purposes of this thesis the product development process will be described up to the validation phase. The purpose of this section is to define the process. Later sections of this thesis will evaluate the process utilizing a risk framework to identify weaknesses in this process that are not being addressed by current process improvements, initiatives, or risk mitigation plans.

3.2.1 Concept Initiation Phase

Identification of a business opportunity initiates the passport process for a new program. Business opportunities are identified when customer dialogue confirms a need for new product requirements that align customer requirements with the company strategy. For new products, the assigned study group establishes a feasible concept to address preliminary system requirements for the business opportunity in accordance with Engineering Standard Work for Concept Initiation. Engineering Standard Work will be defined in section 3.4.2. Any concept is reviewed at a system level product review, to confirm the feasibility of the selected concept and its ability to meet initial system requirements with projected technology readiness. The concept initiation process results in the creation of the initial requirements documentation by the three Systems Engineering organizations. The Executive Committee defines the highest level task by initiating a Job Ticket. The Job Ticket is the basis for venture and program assessment to track progress and risk/gap closure plans.

3.2.2 Concept Optimization

During Concept Optimization, the designated program manager or existing program IPMT finalizes the program Job Ticket. The finalized Job Ticket defines project objectives, compares current capability to the desired result, identifies customer requirements, success factors and measurements to be used to assess product performance, program schedule, and earned value accomplishment. It identifies and provides the strategies for execution of the program and subsequent support of the product, resource needs, management reserve levels, and a winning solution for the system application. During Concept Optimization the engine system concept is optimized by the designated design group to maximize customer value within projected technology readiness and in accordance with Engineering Standard Work for Concept Optimization. The concept of technology readiness will be discussed relative to technology risk to the product development process later in the thesis. The requirement documentation describes the overall propulsion system requirements relative to the vehicle or powerplant defined in the venture analysis and includes:

- Program resource requirements including Level 1 Program schedules, budgets, and staffing
- Source plans
- Verification requirements
- Aircraft or Power Systems Interfaces and System Interface Control Documentation
- Concept defined capable of meeting system requirements
- Updated high level requirements
- Technology Readiness Plan established to achieve TRL 6 readiness by completion of Preliminary Design
- Long lead Industrial Plan formulated
- Program risks identified

3.2.3 Preliminary Design Phase

During the Preliminary Design Phase of a program the IPMT finalizes the requirements documentation for Detail Design, confirms capability to achieve technology readiness (TRL=6), defines the bill of material (BOM) for development engines, and establishes the engine validation and risk mitigation plans. The IPMT also defines the baseline Earned Value Management System (EVMS) plan for detailed design, development, and validation efforts. The EVMS process will be discussed in section 3.4.1. During the Preliminary Design phase the component and part requirements are developed. The Component Integrated Product Teams(CIPT's) and Integrated Product Teams(IPT's) mutually develop the part requirements and the component requirements throughout the preliminary design phase into its final state for presentation at Passport Review PII. The mutual development is enabled by the modular nature of the engine. The part requirement document provides a record of model specific part requirements. The part requirements include part delivery schedules and process and quality assurance plans developed by the CIPT's and IPT's. The source plan for the program is also established within the IPMT. At the start of the first step of Preliminary Design, the draft requirement documentation and an initial Engine validation schedule are provided by ISE to the CIPT's and Partners. In the first step of Preliminary Design:

- The CIPT's are organized
- IPT's are formed
- Standard work and technology readiness are assessed
- Risk assessments and corresponding mitigation plans are incorporated into the program plan
- Design definition from Conceptual Design is reviewed against the draft requirements documentation with gaps identified
- ISE addresses identified gaps and re-allocates component requirements, as necessary, to maintain systems level requirements
- Necessary planning activities are executed
- Technical Performance Measures (TPM) and success criteria at the systems level are established consistent with requirements

The deliverables of the preliminary design process are listed as follows:

- Engine system preliminary design capable of meeting system requirements
- Technology Readiness established
- Finalized requirements documentation
- Preliminary Industrial Plan for hardware delivery defined
- Risk Mitigation Plan established
- Engine validation schedule finalized
- Development Bill-of-Material established
- Configuration Control Board established
- Baseline EVMS plan

3.2.4 Detailed Design Phase

Following approval of the PSRD, and the Preliminary Design definition by the Executive Committee at Passport Review II, the IPMT is authorized to initiate Detailed Design. The Detailed Design Phase consists of the analysis, design and the required rig and engine testing necessary to define the production configuration with sufficient certainty to progress to the Validation/Verification Phase. The design definition from Preliminary Design is reviewed against the final requirement documentation and necessary modifications to the requirements are made to ensure alignment and closure of requirements with the Preliminary Design definition. The Systems Engineers and CIPT's then drive any revisions to the requirement documentation to the supporting IPT's. The IPT's develop the detailed design definition consistent with the analytical and experimental rig and engine testing as prescribed in Engineering Standard Work. The Bill-of-Material resulting from the analytical and experimental design activity is reviewed to confirm requirements are met. CCB approval confirms closure against those requirements. System and Passport III reviews are conducted to complete the Detail design phase of the product development process. The Detailed Design Phase Exit Criteria and Deliverables are listed as follows:

- Component design definition including parts
- Manufacturing part requirements (cost, schedule, process)
- Refined source plan including production and aftermarket requirements
- CIPT risk assessment and mitigation plans for validation and delivery
- Refined EVMS baseline plan through certification/qualification/commissioning

- Validation schedule incorporating CIPT/IPT requirements
- Preliminary design of support equipment
- Assembly and manufacturing tooling requirements
- Initial test results from experimental rigs and engines
- Refined Industrial planning incorporating key supplier input

3.2.5 Verification and Validation Phase

This phase consists of the development and planning activity required qualifying the product for service and establishing production and service readiness. The verification and validation elements within this phase ensure the system, as designed and built, meets all established requirements and customer needs. A System verification plan and its associated engine validation schedule document the requirements of product development, verification and validation. These plans consider and include certification/qualification and/or commissioning tests as required. The verification and validation process establishes the methodology to assure test, demonstration, and inspection requirements are achieved. The verification plan and validation schedule include all test activity at part, component, and system levels, as well as any required Air System/Air Vehicle/Product qualification and certification testing. The delivery elements matured within this phase establish the critical production and aftermarket support process requirements for which build plans and production schedules are defined and verified. Design of the final product is optimized through the verification and validation phase at the part, module and engine system level by assessing part, module and system performance, ease of manufacture, supportability, repair capability, and cost of the development hardware and the testing. The end point of the process is commercial engine Type Certification or military IFR. For new products, the certification/qualification requirements to initiate commercial Air Vehicle or military Weapon System development flight testing concurrently with the validation of the production engine bill-of-material are established by FAR 33 for commercial applications or by the IFR requirements in the applicable military specification.

Verification and Validation Phase Deliverables and Exit Criteria

- Successful verification of identified requirements at the Level 3 Product Reviews (part, module and system)
- Certification or Qualification testing completed in accordance with the appropriate regulatory agency requirements, including but not limited to FAA FAR33 or approved Military Specification
- A Production Readiness risk assessment
- Field Support and Assembly tools, procedures, training and training aids. Field manuals shall be prepared and validated in concert with the engine qualification and certification testing.
- Validation of system requirements to customer needs
- Production assembly and test instructions
- Production source approvals
- Technical publications, support equipment, support tooling, and associated training
- Post certification risk mitigation plans
- Build plans and production schedule
- Integrated support and aftermarket plan
- Repair strategy and initial field repairs developed
- Repair sources defined

3.3 Organizational Structure at Pratt & Whitney

The purpose of this section is to define the organization that has evolved to reduce the risks introduced by the complexity of the product architecture. The risk framework reveals that communication, coordination, and integration quality is key to reducing risk. The organization provides the means to drive high quality communication, coordination, and integration, which will reduce uncertainty in the process.

From the inception of the company to 1990 Pratt & Whitney took the form of a functional organization. Engineering, manufacturing, and customer support were separate functional organizations. The organization required a high degree of technical knowledge in many functional disciplines to produce the highly complex product. The product was in the early stages of the product life cycle. At this point the need at the early stage of development was driven by technical improvements. High levels of core engineering knowledge were required to drive technical improvements. As the product moved closer to a mature product the product need moved toward a broader vision of requirements. The product not only had to be made better; it had to be developed faster and made cheaper. During the 1990's Pratt & Whitney's organizational structure began to evolve into the current structure in a drive to architect the product while taking into account the entire product lifecycle. In 1990, during the early development of the PW4084 and the F119 EMD engines the Integrated Product Development process was instituted and the matrix organizational structure was employed. Component centers were formed around the modularity of the engine and co-located to develop the engine. Component centers were cross-functional organizations with manufacturing and customer support represented. The next evolutionary step, begun in 1995 was to form Product centers that were initiated to bring manufacturing and engineering even closer together. Manufacturing issues were not considered until late in the product development process and costs were not minimized for the product. The product centers co-located the engineering function at the manufacturing site. Finalization of this step occurred when all military development was moved from West Palm Beach, Florida to the various manufacturing sites in the Northeast during the late 90's and early 00's. The co-location of military and commercial products would bring more consistent product architecture to all of Pratt & Whitney's products. The implementation of the product centers resulted in the fragmentation of engineering communications during the development process. In the late 90's the Systems organizations were put in place to handle the coupling issues between the modules and reconnect the organization. The system organization was divided into three organizations; System Design &Component Integration, Propulsion Systems Analysis, Systems Engineering - Validation. These three organizations were equally tasked with the development of the engine and the "three legged stool" made the systemic decisions jointly. By 2000 it was realized that the committee decision making process was extremely difficult and a single chief engineer was appointed to lead the systems chiefs of the three-legged stool. This structure is shown in Figure

3-4For a more complete description of the evolution of the organizational structure at Pratt & Whitney the reader is directed to theses by Mascoli[17], Glynn and Pelland[18], and Bartkowski[19].

3.3.1 Current Organizational Structure

Integrated Program Deployment teams support products and services at the direction of the Program Manager who provides overall business requirements. As shown below, an Integrated Program Management Team (IPMT) is created by the P&W Executive Council, and leads a number of component level teams (CIPT's), which, in turn, lead detail part design and validation teams (IPT's).

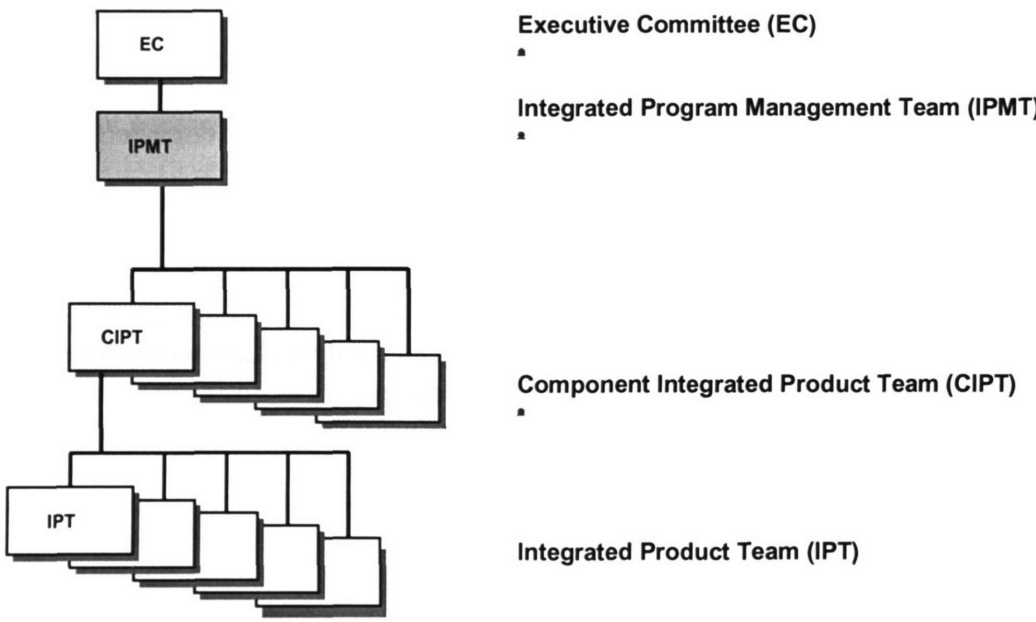

Figure 3-3 : IPD Organizational Structure

The IPMT is a senior level management team that is responsible for managing the project and ensuring customer requirements are satisfied. The Program Vice-President leads the IPMT and has representation of from the groups shown in Figure 3-5. The IPMT is responsible for overall management of the Product Development process for each product. The IPMT provides direction, budget, scheduling, systems level requirements, and organizational goals to the CIPT's and

[17] Mascoli, Greg, "",

Manufacturing organizations.

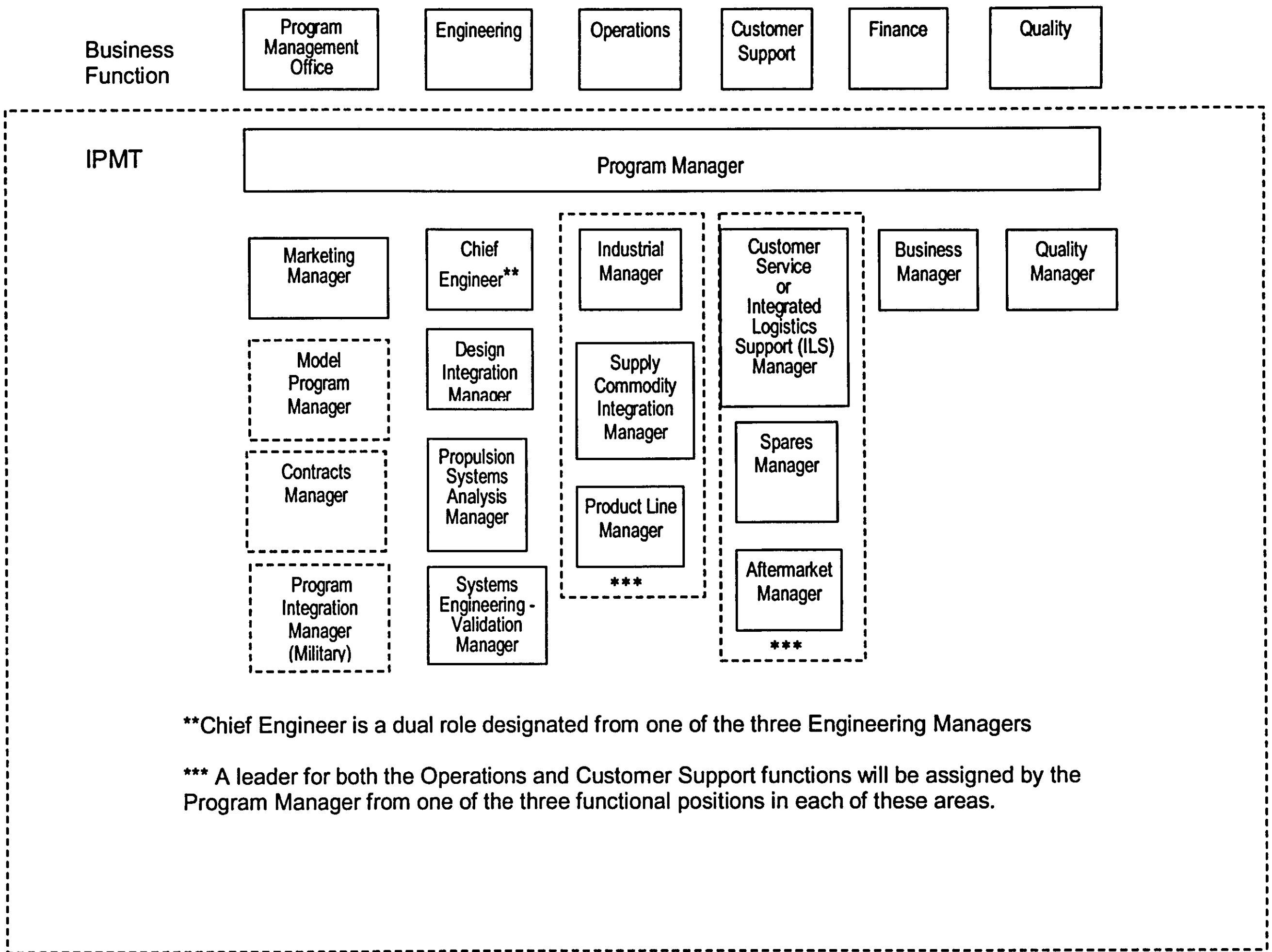

Figure 3-4 : Integrated Program Management Team (IPMT) Structure

The Program Office Business Functions on the IPMT consist of the Program Manager, Model Program Manager (when appropriate) and the Marketing Manager. These functions are responsible for ensuring the IPD process is followed to provide the customer with the product that they require. Resources (budget, staffing, and schedule) required to accomplish the goals are allocated within the given constraints. Program level metrics are established and tracked. The Engineering positions on the IPMT consist of the Chief Engineer, which is a dual role assumed by one of the three Engineering Managers and appointed by the Vice President of Engineering with concurrence of Program

[18] Glynn, and Pelland, "",

Manager. The Engineering functions include the Design Integration Manager, the Propulsion Systems Analysis Manager, and the Engine Validation Manager. The engineering functions are responsible for ensuring technical goals are met within budget and on schedule. The systems engineers lead all product development efforts to ensure systems level requirements are met. The product configuration is managed by the systems engineering organizations. System functional requirements are defined that meet customer requirements. The systems organizations also own and implement the product verification and validation process. The operations elements on the IPMT include the Industrial Manager, the Commodity Integration Manager and the Product Line Manager. The Program Manager will identify and assign one of these positions as the lead Operations role for the program. The operations elements are responsible for the supply chain. They ensure the suppliers are procured. Hardware delivery is managed through management of technology insertion and risk mitigation. The hardware must be managed through the entire lifecycle and procured at lowest cost. They also own the MRP plan. The customer support business function elements on the IPMT include the Customer Service Manager, the Spares Manager, and the Aftermarket Manager. The Program Manager will identify and assign one of these positions as the lead Customer Support role for the program. Customer service establishes the Support Systems requirements for product supportability and maintainability. The finance representative on the IPMT is the Business Manager. The Business Manager constructs and maintains IPMT venture analysis, business cases, and what if scenarios. They develop program financial metrics, track status of program financial performance, and forecast estimate at completion. Overall ownership of the profit and loss statement and business case is within the scope of the business manager. The quality representative on the IPMT is the Quality Manager. The quality manager owns the program passport process and monitors quality metrics. The above structure allows the management of the program to cross all functions of the organization to provide a basis to evaluate all strategic priorities of the various functions. Roles and

[19] Bartkowski, Glenn, "",

responsibilities have been defined for each function within the IPMT through the use of the systems level procedures.

The CIPT leads the IPT's within a given module and engine program to ensure all program level objectives are met. The CIPT is responsible for the integration of all their respective IPT efforts at the system level. The CIPT interfaces with the systems engineering organizations to manage the component system boundaries. Management of the boundaries ensures the total engine system requirements are met or at least not adversely affected by component system changes. A CIPT exists for each identified decomposed element (component) of the engine; Fan, Low Pressure Compressor, High Pressure Compressor, Diffuser/Combustor, High Pressure Turbine, Low Pressure Turbine, Externals, Controls, and Mechanical Systems. The Module CIPT's and detail IPT's work together to manage the execution of the program at the module and detail part level. They are responsible for the following plans: technology, recurring costs, engineering and development costs, design-make, and supplier integration. The Module CIPT's and detail IPT's must also provide Level II and III schedules, resource deployment plans, and conformity validation plans. The organizational structure of the CIPT and IPT organizations is shown in the figures below.

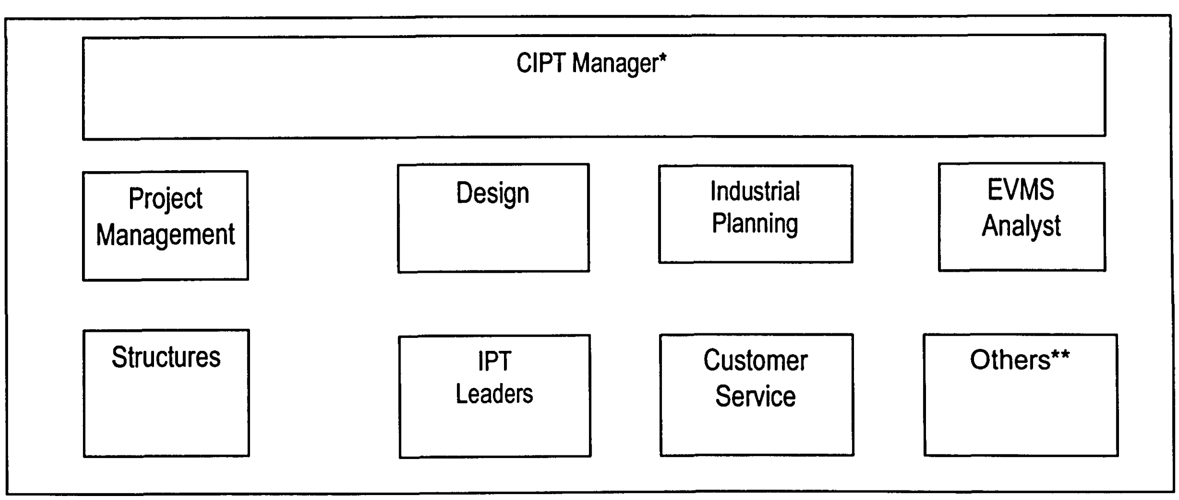

Figure 3-5 :Component Integrated Product Team (CIPT) Structure

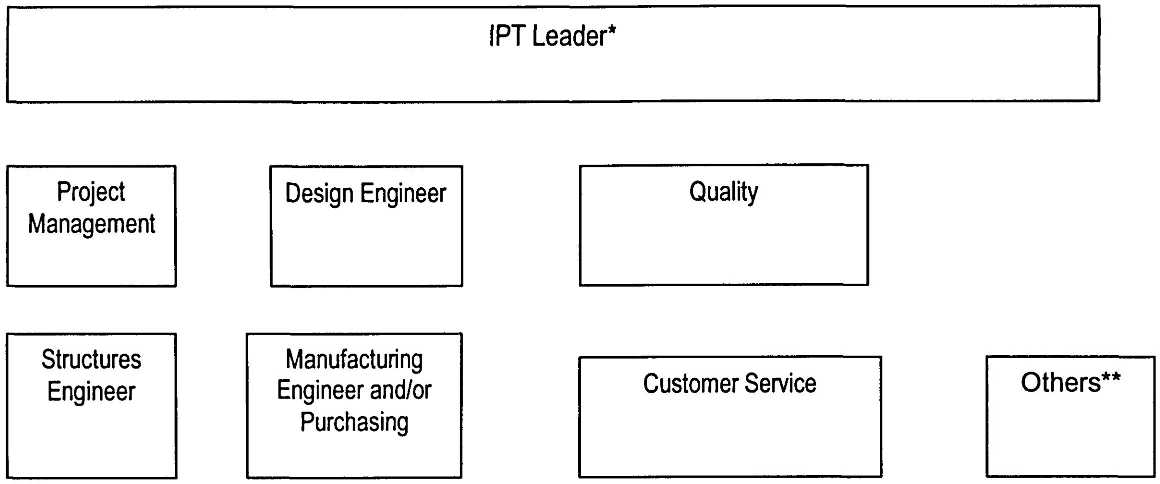

Figure 3-6 : Integrated Product Team (IPT) Structure

3.3.2 Summary

The organizational structure at Pratt & Whitney has evolved quickly over the last decade or so. After many years of operating in a functional organization the company began to move toward a matrix organization. The major strengths of the matrix organization being clear customer interface, rapid reaction, reduction in duplication, and easily disbanded[20]. Over the years the organization took advantage of collocation to enhance engineering and manufacturing communication and take advantage of more commonality in component and subsystem development. Systems organizations have become stronger to enhance the Integrated Product Team structure. Forsberg, Mooz, and Cotterman state that the role of systems engineering becomes crucial when integrating a system developed by multiple product teams.[21] Finally, the roles and responsibilities of the organizational structure are being clearly defined to help with the complex management of the matrix organization.

[20] Forsberg, Kevin, Mooz, Hal, and Cotterman, Howard, "Visualizing Project Management, Second Edition", John Wiley and Sons, Inc, (2000) (pp140)
[21] Forsberg, Kevin, Mooz, Hal, and Cotterman, Howard, "Visualizing Project Management, Second Edition", John Wiley and Sons, Inc, (2000)

3.4 System and Project Management Tools

3.4.1 Earned Value Management System (EVMS)

Cost attentiveness, available budget, and quality of budget planning are all key drivers of risk for cost and schedule. At Pratt & Whitney the Earned Value Management System has been implemented to provide Pratt & Whitney program management with the information, tools, and methods to effectively manage programs and fulfill contractual cost and schedule requirements. The risks relative to cost and schedule defined above can effectively be managed using the EVMS process. Management information systems integrate the planning, scheduling, budgeting, work authorization, and cost accumulation. With this information, the IPMT for each program is able to asses program status, evaluate performance, analyze problems, and implement corrective action in a timely manner. Pratt & Whitney has implemented SAP's Enterprise Resource Planning system in an effort to make the data available more real time and speed the decision making process. The system was implemented across the entire engineering organization during the fall of 2002. Work at Pratt & Whitney is organized and defined within the framework of a Work Breakdown Structure (WBS) by the IPMT. The WBS allows for work assignment, budgeting, scheduling, risk assessment, cost collection, and performance reporting.[22] Systems level elements of the WBS are responsible for the technical performance, schedule, and cost value propositions of the components. Component level elements of the WBS are assigned to the CIPT's who are responsible for the value propositions of the components. The part or task level of the WBS is delegated to the IPT's. The WBS breakdown is shown in the table below.

[22] Ibid

FAN
LPC
HPC
Burner/Diffuser
HPT
LPT
Mechanical Components (Brgs, Seals, Drives + Access)
Controls
Nacelle
PSA (Logic)/FADEC
PSA (Operability/Performance)
MPE (Materials & Processes Eng)
Mid Thrust Prop Ctr - Model Mgmt
Sys Eng - Validation
Sys Design and Cmpt Integ
Tech Support
PMC
Engine Services
Support Equipment Operations
Technical Pubs and Maint Serve

Table 3-1 : Typical Program Work Breakdown Structure

The WBS follows the engine decomposition and mirrors the CIPT organizational structure. The EVMS process provides the status of the program based on reality versus the planned cost and schedule. Problems identified by variances require corrective action quickly if plan metrics are to be maintained. The EVMS compares the budgeted cost for work scheduled (BCWS), budgeted cost for work performed (BCWP), actual cost for work performed (ACWP), budget at completion (BAC), and estimate at completion (EAC). The data is summarized for each WBS. This information allows management to monitor performance, analyze variances, assess the known work remaining, implement corrective action, and report status throughout all phases of a program. Variances are defined and monitored for cost and schedule.

$$CostVariance = BCWP - ACWP$$

$$CostVariance\% = \left[{(BCWP - ACWP)}/{BCWP} \right] * 100$$

$$ScheduleVariance = BCWP - BCWS$$

$$ScheduleVariance\% = \left[{(BCWP - BCWS)} \big/ {BCWS} \right] * 100$$

Positive variances indicate cost underruns or ahead of schedule conditions. Negative variances indicate cost overruns or behind schedule conditions. The IPMT establishes variance analysis thresholds considering such factors such as customer requirements, past experience, IPD risk assessments, and high value items. Variances that exceed established thresholds must be explained and corrective action planned. The thresholds are sufficiently restrictive to ensure timely identification of evolving problems in cost and schedule performance. The EVMS system is designed to handle changes in budget, schedule, and scope of work by segregating the changes into two categories.

- Out of Scope Changes (those attributed to Customer or Contractual Changes)
- In-Scope Changes (Internal Replanning) which includes:
  - Out of Plan Changes (Rework)
  - Future work (Additional work to unopened work resulting from reapplication of resources to improve schedule)
  - Incomplete work (Unrealistic plans requiring re-planning)

Pratt & Whitney has used the EVMS process for managing military programs since the middle 90's. The process is being implemented in the commercial programs this year. Management of all programs will be standardized and performance will be able to be monitored. The EVMS system will help identify problems early during the IPD process. Once the problems are identified Pratt & Whitney utilizes the quality initiative known as Achieving Competitive Excellence (ACE) to fix problems completely and correctly the first time.

3.4.2 Quality Management

Risk drivers related to Quality are design concept initial quality, communication, coordination, and integration quality, budget planning quality, amount of design mistakes, and problems related to unknown, unknowns. This section defines the company practices related to quality. The following section will define the risk drivers and relate them to the background presented here.

Pratt & Whitney's Quality Policy states that Pratt & Whitney is committed to being the world class provider of dependable engines, propulsion systems, parts and services that meet customer expectations. The diagram below illustrates the relation ship between the core IPD business process and the initiatives used to maintain and improve quality.

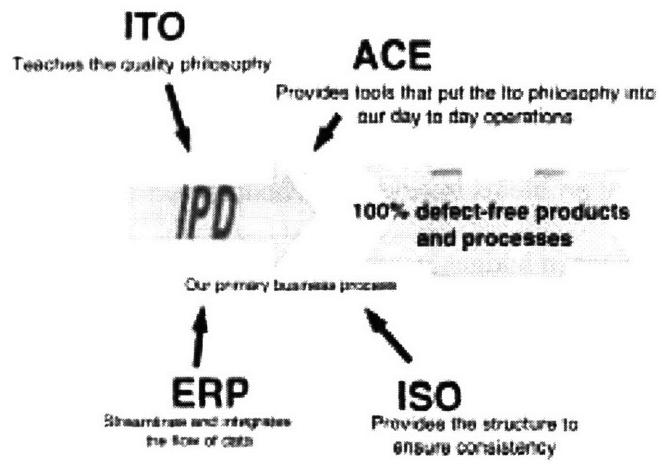

Figure 3-7 : Pratt & Whitney Quality Philosophy

The first two items to note in Figure 3-7 are ITO and ACE, which are two United Technologies initiatives aimed at continuously improving the quality of the company's processes. Ito University is the internal management-training program, which teaches the philosophy behind the ACE program. Ito University was established in 1998 with the purpose of bringing the proven techniques and principles of Mr. Yuzuru Ito, former advisor on quality, to all employees of UTC worldwide. ACE, which stands for Achieving Competitive Excellence, is an effort to incorporate quality into everything the company does. The ACE program includes a comprehensive system of tools and metrics, common across all of UTC, which are used to continuously improve both manufacturing and business processes. The goals of the ACE program are to involve and empower front-line leadership and the workforce to be owners of their process, to reduce the cost of poor quality, to sustain continuous improvement in process efficiency, to improve competitiveness and to align all resources under one integrated initiative.

The ACE initiative uses market feedback to assess customer satisfaction with the value delivered and identifies the required areas of improvement. The passport review process monitors the progress of the programs toward delivering the product to the customer and measures the performance gaps relative to the customer requirements. The performance gaps can be analyzed using quality tools that analyze the opportunities for improvement by identifying the inefficiencies in the process. The inefficiencies are defined as turnbacks. The ACE initiative also identifies seven elements that may be used to improve the process and reduce the performance gaps. The elements can be used when opportunities for improvement are identified or are continuous improvement tools. These elements are 6S(Sort, Straighten, Shine, Standardize, Sustain, and Safety), Total Productive Maintenance (TPM), Relentless Root Cause Analysis (RRCA), Mistake Proofing, Process Certification, Set-up Reduction, and Standard Work. 6S is used to create an atmosphere for continuous improvement. TPM involves all employees in achieving maximum equipment effectiveness by achieving less machine downtime. RRCA identifies the root cause of failures within the systems. Mistake Proofing refers to the idea of achieving a 100% defect free process. Process Certification and Set-up Reduction involve lean practices to remove waste from the processes. The most sweeping element of the ACE initiative is the element of Standard Work.

A definition of standard work at Pratt & Whitney is:

A disciplined approach to achieve business process effectiveness, efficiency, and agility. Standard work is a method for capturing both process and product knowledge. Standard work provides a prescriptive documented process supported by consistent repeatable instructions, and a method for recording results. It relates the best process approach to date and assesses historic levels of performance (capability) to frame the expected results.

Elements of Standard Work

To adequately provide instructions for the product development process and provide the requirements of the product development process the standard work defines the following tools. Process flow maps achieve the goal of defining the activities required to execute the product development process. Documentation is generated for each activity that is defined in the process flow maps that provides summary information and detailed work information. There is also documentation, which provides the design criteria that must be satisfied by the final product. Finally design standards are documented that defines preferred configurations for systems, modules, and parts that are aligned with manufacturing processes and design lessons learned. All deviations to standard work processes are also required to be documented.

The reorganization of the company and subsequent application of standard work resulted in sporadic use throughout the company with the most consistent use at the CIPT and IPT levels. Incentives were not aligned to drive change throughout the company.[23] Management has responded by adding conformance requirements to the process that requires identification of the process map and activities used to complete every task. The product review process verifies the conformance. These processes will add substantial work that the program staff must complete in addition to the work that is required to design the product. The process will enhance the ability of individuals new to the company to learn the process much faster if the requirements are documented rather than relying on tacit knowledge transfer. The definition of the process allows areas that are similar between projects to be defined easing the transition of employees from project to project. The effect of productivity loss due

[23] Bartkowski, Glenn, "Accounting for System Level Interactions in Knowledge Management Initiatives", SDM Masters Thesis, (February 2001)

to context switching will be lessened. Finally process management techniques to identify waste and identify bottlenecks may be applied to drive higher productivity if the tasks to perform a process are fully defined.

A few specialized ACE teams have also been formed to attack high level issues that directly affect customer satisfaction. These teams are involved in projects such as achieving program cost goals, first design quality, product cost, and maintenance cost. Later sections will elaborate on how these initiatives address key risk drivers in the product development process.

3.5 Summary

The purpose of this section is to define product architecture, organizations, and processes required for developing gas turbine engines. The definition will be related to risk drivers in the next section. The evolution of the organization, development and definition of the processes, and institution of the quality philosophy are designed to reduce the risk levels for performing the product development process within the cost and schedule constraints. The relation of the background section to the key risk drivers will enable the thesis to find the root causes of cost and schedule uncertainty that are not addressed by the above-defined system. In addition key drivers that the system is attempting to improve, which are quality and productivity, will be evaluated in the policies that are defined with system dynamics model.

The level of complexity effects the amount of risk a program takes in achieving cost, schedule, and performance goals. The complexity is determined by the system of systems that are involved in the endeavor and are defined by three systems. The systems are Product (Architecture), Process, and Organization. The architecture of the gas turbine is both modular and highly coupled. The coupling of the architecture drives much higher complexity into the system. The complexity introduced by the architecture drives the need for a highly developed organization. The organization has taken over a decade to evolve to adequately handle the product development process and is continuing to evolve. The definition of the product development process for the gas turbine engine details high levels of requirement definition and planning. A highly defined process also enhances the ability to control the process. For such a complex product control of the process is key to implementing process improvement initiatives. Definition of risks and identification of risk mitigation plans in the complex product development process is crucial to reducing the cost and schedule uncertainty. Project visibility and control are critical to providing the opportunity to identify variances early. Early identification of variances allows the quality tools to be brought to bear on problems to eliminate them from the process.

4 Identification of Risk Drivers within the Product Development Process at Pratt & Whitney

To define the system performance relative to delivering the product within the cost and schedule constraints of the current product development process at Pratt & Whitney a means of relating the process to a cost and schedule risk standpoint needed to be found. The above-defined system at Pratt & Whitney will be compared to the risk framework developed by Tyson Browning to identify the policies that will mitigate risk to cost and schedule.

The drive of the aerospace industry to become better, faster, and cheaper has elevated the uncertainty of achieving the cost and schedule goals that have been promised to the customer. The literature review provided a causal framework for risk drivers in a complex product development process authored by Tyson Browning. Browning stated that complex system product development involves risk. The risk stems from uncertainty regarding product performance in the marketplace and the ability of the development program to deliver that product within a given schedule and budget - and the consequences of the undesirable outcomes. Browning presented six categories of product development risk. The categories are defined in the table below.

Performance Risk	Uncertainty in the ability of a design to meet desired quality criteria (along any one or more dimensions of merit, including price and timing) and the consequences thereof
Schedule Risk	Uncertainty in the ability of a project to develop an acceptable design (i.e., to sufficiently reduce performance risk) within a span of time and the consequences thereof
Development Cost Risk	Uncertainty in the ability of a project to develop an acceptable design (i.e. to sufficiently reduce performance risk) within a given budget and the consequences thereof
Technology Risk	A subset of performance risk: uncertainty in the capability of technology to provide performance benefits (within cost and/or schedule expectations) and the consequences thereof
Market Risk	Uncertainty in the anticipated utility or value to the market of the chosen "design to" specifications (including price and timing) and the consequences thereof
Business Risk	Uncertainty in political, economic, labor, societal, or other factors in the business environment and the consequences thereof

Table 4-1 : Categories of Product Development Risk (Browning, Tyson)

I will address each of these drivers in relation to the background to identify areas within the company that are do not have adequate risk mitigation plans. Effective risk management of the system requires continuous monitoring of project risks and effective control mechanisms for identifying and reacting to system instabilities. Browning does point out that, without a systems view many risk management actions serve only to push risk into another category. With this in mind the risk drivers need to be evaluated to identify if they are risks that can be mitigated or are they so constrained that they push risk into other categories.

4.1 Development Cost Uncertainty

According to Table 4-1 development cost risk is the uncertainty in the ability of a project to develop an acceptable design within a given budget. In the case of all past programs at Pratt & Whitney development costs have overrun original planned budgets. The development cost uncertainty is high given the amount of cash that is required to complete the IPD process. Evaluation of factors that affect development cost will be key to evaluating the product development process for weaknesses in the system. See Figure 4-1**Error! Reference source not found.** for the causal framework for factors that contribute to Development Cost Uncertainty.

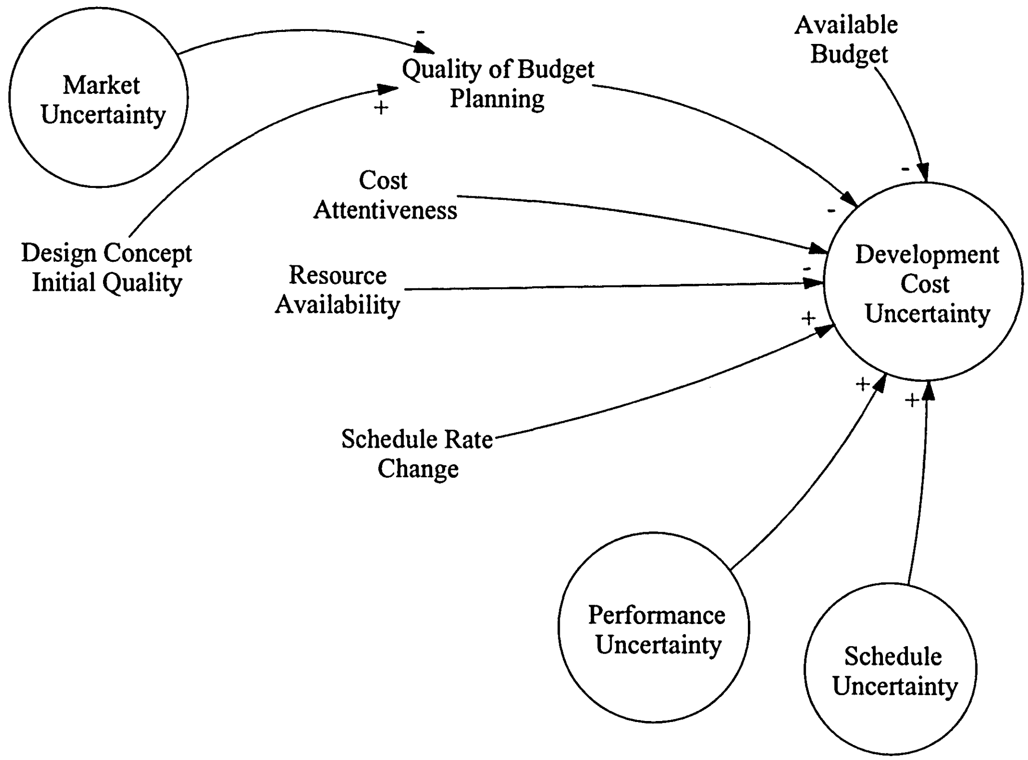

Figure 4-1 : Factors Contributing to Development Cost Uncertainty

As can be seen by the causal diagram, performance uncertainty affects development cost uncertainty. The ability of the program to mitigate performance risk directly affects the need for the program to incur more cost to accommodate performance shortfalls. In section 3.1 4.1 the thesis addressed performance uncertainty and the fact that Pratt is working on many risk mitigation processes. Performance risk is inherently high due to the nature of the product architecture, but a great deal of effort is being expended on driving the risk down. The other uncertainty that affects development cost uncertainty is schedule risk. Schedule risk causes the program to last longer than initially planned. If resources are held constant and the program stretches longer than planned higher costs will be incurred due to carrying resources. Cost is usually even more affected by schedule slips because resources are normally added to a late program to keep to the original schedule. Therefore schedule uncertainty is a significant factor affecting cost uncertainty. The thesis will explore drivers of schedule uncertainty in the next section. The following subsections will explore Pratt & Whitney's current state in relation to the development cost risk drivers.

4.1.1 Cost Attentiveness

Program managers require the ability to understand and monitor project costs. Real time feedback on costs allows faster ability to find and fix the problem that caused cost to deviate from the plan. The ability to apply corrective action to cost variances from the plan quickly can lower cost uncertainty. Pratt & Whitney instituted a company-wide program to monitor product and company financials. Section 3.4.1 on the Earned Value Management System (EVMS) as instituted at the company outlines the new system.

4.1.2 Available Budget

Available budget is an outcome of business and market risk. Corporate financial health affects the available budget. The events of September 11th, 2001 affected the available budget for development programs. The current turmoil of the airline industry affects the cash flow of the company and the available budget for programs. On the military side the current administration's attitude toward military budgets affects the budget allocated for development programs. If the budget is inadequate to cover all planned activities and a certain amount of unplanned activities staffing will be inadequate staff the programs to cover the requirements. A vicious cycle due to low staffing can be initiated where overtime causes fatigue and high rework. The added work increases cost to the program. Development cost uncertainty increases with the amount of additional rework. As stated before all recent major development programs such as the ones defined in this thesis always overrun the allocated budget. Work that was not planned had to be performed to complete previous programs.

4.1.3 Quality of Budget Planning

The quality of budget planning will affect development cost uncertainty in that well planned programs will have lower risk than poorly planned programs. Poorly planned programs will not have adequate budget to cover the planned tasks. The scenario presented above will be the rule. In the commercial programs at the company the situation normally arises where the budgets get planned and must be adjusted to meet the available budget. Since budgets are low relative to the program requirements, the quality of the planning suffers as work to do is lowered to meet available budget.

4.1.4 Resource Availability

High resource availability lowers the cost uncertainty by increasing the ability of the program to handle unintentional iterations and to utilize flexibility to re-plan the programs. Resources being available when required so that work can be done in a timely manner affects the cost of a product development program in two ways. If resources become scarce bottlenecks will occur in the process. Bottlenecks lead to resources being idled because work that is dependent on other work being completed can not be completed. The program will be paying for those idled resources. Also, schedule will suffer if resources can not be brought in to expedite when unplanned work is greater than the resources can handle. Schedule will slip and costs will rise because the resources will be generating costs longer than planned. Resources can be property, plant, equipment, and personnel. Personnel must also be available in the right quantities in respect to expertise and experience. The organizational structure of Pratt & Whitney is designed to minimize the cost of paying for idled personnel as they can be assigned work from other programs to fill the void. Moving people from one program to another quickly has presented problems. These problems are most evident when the programs have fallen into a firefighting mode and the program becomes severely understaffed. At these times it is particularly difficult to move personnel between programs. The ability to ramp up resources is diminished so projects are forced to wait for resources. The programs become delayed and staffing can be idled waiting for resources to free up. Resource availability is a major source uncertainty within the process at Pratt & Whitney. The system dynamic modeling process used to evaluate the ability to lower cost and schedule risk will have to center on resource availability. Methods that have been identified through experience and literature review are overtime, utilization of outsource labor, and lower utilization of staff to enhance flexibility.

4.1.5 Schedule Rate Change

Schedule Rate Change refers to accelerating or decelerating the rate of doing work in a product development program. As identified above lower utilization of staff will enhance flexibility and the ability to accelerate the rate of doing work. The causal framework indicates that performance uncertainty, schedule uncertainty, and firmness of the deadline affect the schedule rate. In the case of the gas turbine engine development programs shortfalls in performance cause unintentional iterations and lengthening of the schedule on planned activities. The firmness of the schedule has been set by the MOU at the beginning of the programs. Penalties are levied against the company for schedule slips in the case of the commercial engine programs and the risks of program cancellation go up in the case of military engine programs. The development programs are always accelerated in the face of any schedule uncertainties to protect the ability to deliver on time. Most programs are intentionally accelerated to keep the sense of urgency high so schedules may be protected. The problem with accelerating the projects is that the practice drives overtime which was not originally planned. Program cost risk is elevated in this case. The company recognizes that acceleration of the planned program causes development cost growth and is attacking the drivers of schedule rate change.

4.2 Schedule Uncertainty

Since the advent of better, faster, cheaper, one of the main goals of companies is the reduction in cycle time of the product development process. The gas turbine engine development process requires a faster process to better match up with the aircraft development cycle. Typically the development time for a new aircraft is shorter than the engine. The difference in development time causes great difficulty in matching definition of requirements. Requirements for the engine can not be defined if the aircraft has not started preliminary design. The industry recognizes this and is attempting to better align the development processes. "In the past, we started working on an engine long before the requirements for its airframe application were set; mostly because engine development cycle times were so much greater than those of the aircraft they were planned to power," Mike Benzakein, GE's general manager of advanced engine programs, said. "That meant that as aircraft requirements evolved, we had to constantly play catch-up. Not having the aircraft and engine development cycles in sync cost us time and money, and sometimes affected engine and aircraft performance."[24] To better align the cycle times means the engine development cycle time must be shortened. The initial level 1 schedules that are agreed to in the MOU for the most recent programs are shorter than development schedules in the past. Schedule uncertainty is raised due to fact that the promised short cycle times have never been achieved with the current processes. The figure below shows the Browning causal diagram for schedule risk.

[24] Kandebo, Stanley, "General Electric Aims at 18 Month Engine", Aviation Week and Space Technology (November 2002)

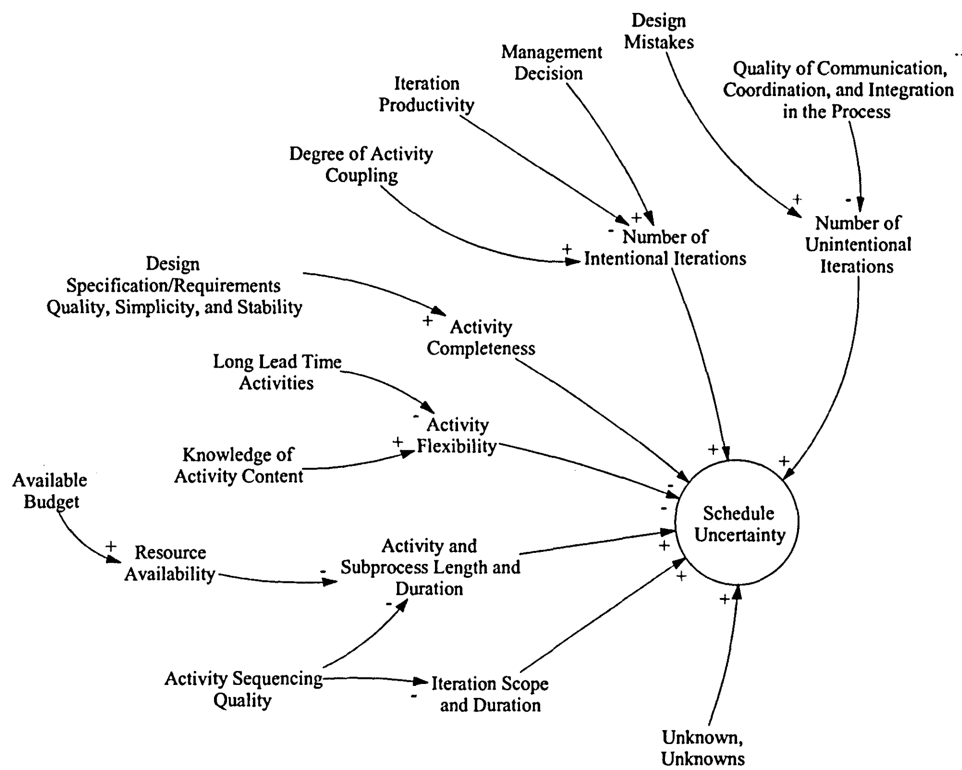

Figure 4-2 : Causal Diagram for Schedule Uncertainty

4.2.1 Work to do (Intentional Iterations) Vs re-work (Unintentional Iterations)

The product development process is a process of doing work. Work can be done correctly or it can be done incorrectly. The factors affecting the amount of work being done correctly are productivity and quality. In the case of the Browning causal diagrams the known tasks that must be completed correctly are labeled intentional iterations. Iterations are used to refine the design during the product development process. Browning labels the work to do in the complex product development processes as an intentional iteration due to the fact that the coupled nature of the tasks will require input from multiple activities. The information exchange can only be accomplished in multiple rounds. The Pratt & Whitney process has been studied in many theses by using Design Structure Matrices DSM to analyze the information exchange. The work confirms the interdependency of the data required by the systems organizations, CIPT's, and IPT's to complete the development process. Thesis work by

Greg Mascoli[25] indicated that during the concept initiation and optimization phases as well as preliminary design phase there is a high degree of interdependency. During detail design intentional iterations is contained mainly within the component organizations but information must still be exchanged at the interfaces of the modules. Once validation begins, there is again a high degree of coupling of information required across the entire organization as the entire system is being validated. The systems organizations are key to managing the information exchange to bring the development program to completion. Within the framework presented intentional iterations, or the baseline work to do, are performed to complete the design and validation process. The number of iterations will be based on the amount of information dependency and exchange. The basic systems dynamic model of a project is shown in Figure 4-3**Error! Reference source not found.**.

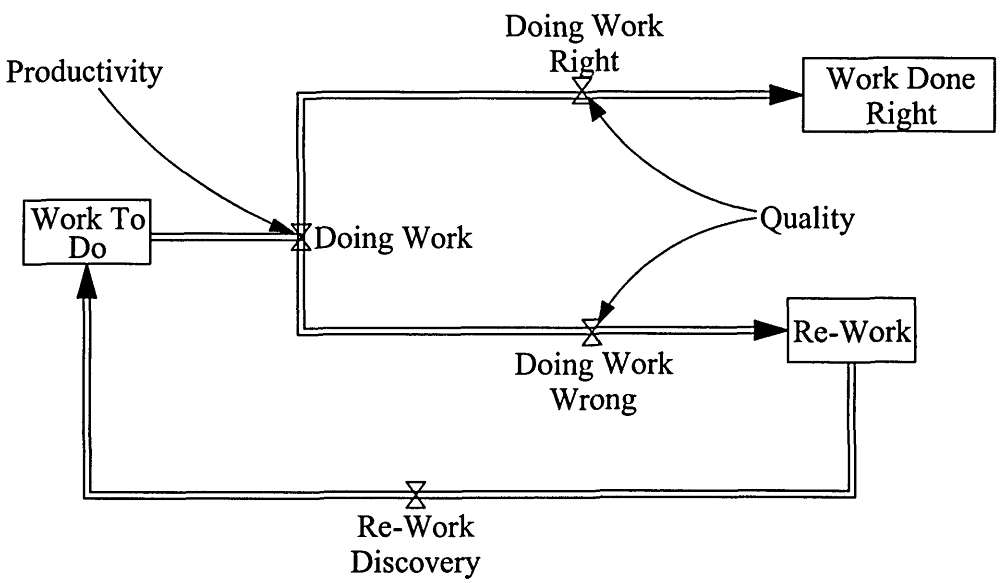

Figure 4-3 : Basic Work To Do Structure

The basic Work To Do systems dynamic model starts with a stock of work that must be accomplished. A stock is an accumulation of things. Over time, work is done at a certain rate that is

[25] Mascoli, Gregory J., "A Systems Engineering Approach to Aero Engine Development in a Highly Distributed Engineering and Manufacturing Environment", SDM Thesis, (January 1999)

affected by productivity of those doing the work. The quality of how well the work is completed affects how much work is done correctly or incorrectly. If the work is done correctly the stock of work to do is reduced and the stock of work done correctly is raised. If the work is done incorrectly there is a stock of re-work that accumulates. The work that was done incorrectly is discovered at a certain rate and winds up adding to the stock of work that must be done. This basic model is the basis of the work to do structure in a multi-project systems dynamic model developed by Greg Herweg and Karl Pilon. The model will be used to evaluate policies that can help mitigate risk in a complex, multi-project, product development process. The model will be further discussed in chapter 5. Relative to schedule uncertainty there are a number of variables that affect the amount of iterations that must be performed. Initial quality of the design concept was discussed relative to technical risk. A tiger team is working on methods of raising the quality of the concept and preliminary design architecture to reduce risk. The discussions of the other variables that can affect the amount of iterations performed are discussed in the following sections.

4.2.1.1 Iteration Productivity

Iteration productivity refers to the effectiveness of the process at reducing the performance gap relative to the customer requirements. To reduce costs and speed the design iterations advanced modeling tools have been employed to the highest extent. The development of high quality modeling tools such as advanced CFD tools, better simulations of the engine performance and control systems, and 3D design tools that allow direct structures analysis reduce the amount of time required to complete iterations. The complexity of the product coupled with the lead-time required to procure hardware during the process requires use of new technologies to conduct complex experiments quickly and cheaply. The company strives to employ the newest technology to enhance iteration productivity and shorten the overall process. The company has focused many efforts upon improving the productivity of the staff in the process. The modeling evaluations will have to assess the ability of these efforts to mitigate the cost risks associated with enhancements in productivity. Again the risk in the product development process is tied to productivity. Productivity must be evaluated for the ability to mitigate cost and schedule risk.

4.2.1.2 Management Decision

From the causal framework above management decision affects the amount of intentional iterations or work to do within the IPD process. As discussed in the background the IPMT is tasked with managing the IPD process. Factors that affect the IPMT decision process on schedule risk are available time and budget. Marketing factors affect how long the product development process can take. Previous discussion identified the fact that airframe manufacturers can usually develop a new aircraft design much faster than an engine product development process. For this reason the available time to meet market demands is shorter than the historical averages to complete the IPD process and the schedule is being squeezed to shorten the time available. Compression of the schedule allows the aircraft system when the airline customer desires the product. The available budget is governed by the business case and controlled by the Executive Committee. History shows that initial allocation of funds for the product development process is not adequate to cover the programs cost. Therefore the management decision has been to plan on lowering the amount of work to do in the IPD process to meet planned cost and schedule levels. Management understands that the decisions they are forced to make due to market and business factors are driving the process to a low number of iterations and are working on improvement of the processes and quality to reduce the schedule uncertainty.

4.2.1.3 Quality of Communication, Coordination, and Integration in the Process

The complex process for developing gas turbine engines demands high levels of communication, coordination, and integration. High quality communication and coordination reduces the potential for mistakes and rework and frees resources that would be saddled with completing the unintentional iterations. In the background section of the paper the product development process and standard work documentation process were documented. The use of a highly defined integrated product development process aids in the information flow process. The evolution of the organizational structure was also discussed. Co-location of the organizations also helps ease the tacit information flow and coordination of the process. The organization and processes at Pratt & Whitney have evolved to mitigate risk to project schedule and continue to be improved. Quality is again pointed out as a major driver to cost and schedule risk.

4.2.1.4 Design Specifications/Requirements Quality, Simplicity, and Stability

Design requirements must be stable and unchanging to avoid rework and define the amount of work that must be accomplished. Browning noted "Gupta and Wilemon (1990) and Mello (1997) found poor definition of requirements to be the number one cause of delays in new product development. Complex and/or unequivocal requirements increase the likelihood that something will be missed on the first pass and cause rework later. Incomplete requirements increase the likelihood that their completion will create new information and rework. Unstable requirements will result in moving design targets and changing prioritization will result in new trade-study results. Ironing out conflicting requirements requires multiple, unintentional iterations."[26] Simplicity of requirements is difficult for the design of such a complex product. Requirement definition is pursued and documented aggressively from concept through detail design as outlined in section 3.2. The desire is to define the highest quality design requirements that can be attained at each of the design stages. There is one document that is defined within the process that contains all of the requirements for design of the gas turbine engine. Having the documentation centralized simplifies the process by enabling the design groups to find the requirements in one place. Some of the dedicated quality tiger teams initiated by the ACE quality initiative aimed at improving the planning process. The process includes the

[26] Browning, Tyson, "Sources of Schedule Risk in Complex System Development", INCOSE paper, (1998)

definition of requirements. Finally the drive to shorten the product development process helps the quality of the requirements definition by enabling the company to obtain better requirements from the aircraft manufacturer. The product development process for the aircraft is shorter than that of the engine. The requirement definition timeline of the aircraft follows the requirement definition for the engine. The shortening of the process will increase the quality of the design requirements. Quality is key to risk mitigation and will be evaluated for the system dynamic effect on cost and schedule uncertainty.

4.2.1.5 Degree of Activity Coupling

High coupling of the activities drives the requirement for more iteration to complete. Modular designs can be developed and tested in parallel with each other without having to consider system level effects. Parallel development of components reduces the number of iterations required since the modules can each go through an iterative design process and they do not have to wait for other components to go through their design process. When high degrees of system coupling is present the system must be tested and go through a systems level iteration process. Module shortfalls resulting from systems level interactions must be evaluated and incorporated in the system. The shortening of the product development has resulted in many of the systems level iterations to be removed. The most telling indication of the desire to remove the system level iterations has been renaming the Product Development and Validation organization to Systems Engineering - Validation. There is a conscious desire to remove the development iteration

4.2.1.6 Design Mistakes

Design mistakes are a result of poor quality in the design process and cause rework. More rework in the product development process increases the schedule uncertainty because more resources or higher productivity is required to maintain the level 1 schedule. The EVMS process is targeted at finding problems, such as design mistakes, as quickly as possible. In addition documentation of the number of problems and the subsequent resolution of the problems are maintained. The data provides the information on the type of problem and the group that owns the problem. An example of the record is shown below. The ACE process tools can be used to find the corrective action to fix the root cause of the design mistakes. The graph shows that problem closure is increasing. Increases in quality will lower the amount of design mistakes. Therefore quality improvements are a key metric that must be evaluated during the system dynamic modeling process.

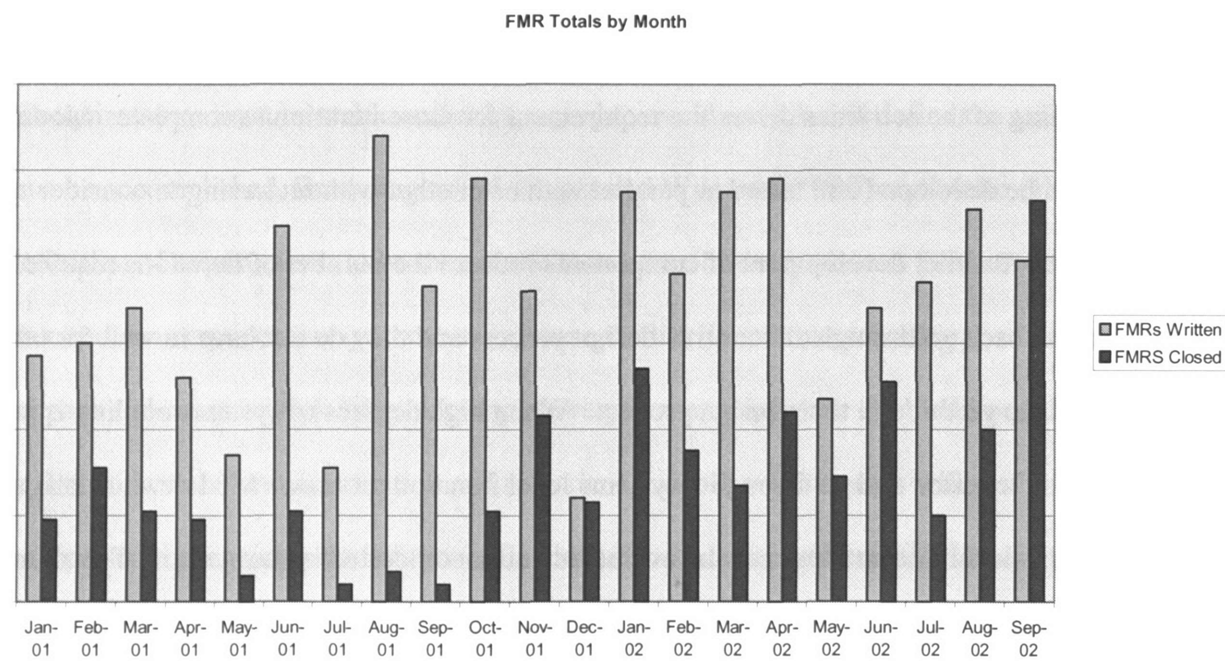

Figure 4-4 : Example Turnback Totals by Month

4.2.2 Activity and Sub-process Length and Variance

The activity and sub-process or task length add up to equal the iteration length and in the end the schedule length. The tasks are variable in length depending on the resource availability and stability, the knowledge of task requirements, and the likelihood the task can be completed successfully. If resources are unavailable or not available for the full time required the tasks take longer to complete. Planned times to complete the tasks will not be correct if all requirements are not fully known. If tasks have a low probability of success, rework will likely be required to complete the task. The likelihood tasks will be completed successfully is defined by the performance uncertainty. The schedule will slip. The product development process at Pratt & Whitney has schedule risk affected due to all of these factors. The available budget drives resources to low levels and resource availability and stability suffer. Projects are re-planned to work around shortages of staffing, tooling, test stands, and test equipment. The requirement definition for the tasks is addressed with standard work. The documentation of the process is intended to outline all of the requirements to complete the tasks. The uncertainty as to the requirements should be substantially reduced, which will reduce schedule uncertainty. The performance uncertainty was discussed in section 4.1.

4.2.3 Activity Flexibility

Activity flexibility or the ability to rearrange the activities in a process allows tasks to continue to be completed when problems arise. If tasks can be rearranged then the risk to schedule is lowered. Since there is a degree of modularity to the gas turbine engine the flexibility of the process is enhanced. Coupled systems begin adding dependencies that make rearrangement more difficult. But experience in the process has shown that even though there is a high degree of coupling the high volume of tasks that must be accomplished for such a complex product there is always an ability to execute a work around strategy. Tasks do not stop when issues such as late hardware due to long lead times occur. The process is rearranged to accommodate the delay. The systems organizations provide the depth of knowledge that is required to evaluate how the problems affect the process and rearrange the validation schedule. According to Browning the lack of agility of the organization to adapting to the new sequence of activities and the amount of long lead activities constrain the flexibility of the process. The company is highly adept at flexibility. During the validation process in all past programs the engine validation plan has been changed multiple times within a weeks time. Problems constantly occur and primary paths to completion of the validation engine must be changed repeatedly. The schedule risk can continually be managed by changing the order of the tasks. The major constraint to activity flexibility is the long lead activities. Hardware redesigns during the process can take 9-12 months before the new hardware is available. Problems of this magnitude occurring late in the process may not be able to be mitigated by activity flexibility. Programs do everything possible with activity flexibility when faced with the constraints to mitigate schedule risk.

4.2.4 Activity Set Completeness

Activity set completeness addresses the issue of clear definition of all activities required to complete the product development process. For the purposes of planning the duration of a project the more the bounds of the project that are defined the less uncertainty there will be in the schedule. As described in section 3.4.2, the use of Standard Work is the process to completely define all tasks required to develop a gas turbine engine. The Standard Work definition is still a fairly new process that is just beginning to completely define the process. The documentation must be continually improved to accurately define all activities. Process improvements can be made to the defined process and enhance the productivity. Defining the entire process reduces schedule uncertainty.

4.2.5 Iteration Scope and Duration

The iteration scope and duration directly effects the schedule risk. Long iterations cause problems when unintentional iterations are required. As the project schedules become shorter the ability to handle unintentional iterations that are long in duration within the promised level 1 schedule becomes impossible. The gas turbine engine contains a great deal of hardware that has long lead times associated with procurement. Hardware can take as long as 2 years from raw material to finished product. The way programs attempt to handle the high uncertainty associated related to this subject is through risk definition and mitigation plans. If risk is high on hardware with a long lead-time associated with redesign and procurement cycles the program will release authorization to procure extra hardware. The added hardware in WIP will only be manufactured to the point where the design risk can still be addressed. In this way schedule risk can be mitigated. Iteration scope and duration can severely affect a program if a design problem is discovered that has no mitigation plan associated with it. Most issues that have large effects on schedule are discovered late in the process and can have severe repercussions on project schedule. The flexibility of the validation process enables some late iteration to be addressed without driving a schedule slip, but they have to be small in scope.

4.2.5.1 Activity Sequencing Quality

As outlined in previous sections the activity sequencing of the product development process for the gas turbine engine is highly complex due to the interdependency of the systems design. Activities or tasks should be sequenced to provide information in a timely manner to activities that require the upstream information. The organizational structure has evolved to incorporate strong systems organizations to aid in communication of the activities. The systems organizations help link the independent component design processes and manage the coupling that exists from a systems perspective. The Standard Work process seeks to fully map the activities and provide clear direction of the sequencing. The definition will allow improvement of the process and elimination of non-productive steps within the sequence. The risk introduced by coupling can be mitigated if the process is defined and streamlined. The number of iterations can be minimized while not increasing schedule uncertainty. Quality is again pointed out as a key metric that drives cost and schedule uncertainty

4.2.6 Communication, Coordination, and Integration Quality

The quality of communication, coordination and integration of information reduces performance uncertainty by providing the information on problems in a timely manner to the rest of the program participants. A great deal of effort at Pratt & Whitney is devoted to information management. All processes are documented as required by ISO 9001 requirements. SAP enterprise resource management software has recently been implemented to consolidate information in one place. Standard work is being written to help understand required work to do on the gas turbine engine product development process. This section shows that enhancement of quality within the communication, coordination, and integration processes will lower the risk of achieving cost and schedule goals.

4.2.7 Unknown Unknowns

All programs suffer from unknown unknowns. The causes of uncertainty that are known can be mitigated. During the IPD process all known technical risks are identified and risk mitigation plans must be identified to reduce the risk. Unknown, unknowns in this process are the failures in the design that were not anticipated. Typically these issues are discovered late in the process and have a great effect on the schedule due to the lead-time required to fix the problem. From lessons learned on various development programs many of these problems are found during endurance testing of the engines. Every effort is made to sequence the endurance testing as early in the process to find the problems before lead-time issues affect the schedule. The tiger team tasked with improving the first design quality is also directed at driving out unknown, unknowns.

4.3 Performance Uncertainty

Cost and schedule risks are the two main drivers of system performance failure in the gas turbine engine product development process. The value delivered to the customer also involves performance risk. The reality of the gas turbine engine business really eliminates the performance risk as a metric that has the ability to be varied to lower the product development process risk. The reason is that the performance is guaranteed to the customer. The product will not be produced unless the performance is demonstrated relative to the promised metrics. As shown in Project Z in the original problem statement the cost and schedule will continue to grow for the product development process until the performance goals are met. This section more defines the immovable nature of the product performance within the gas turbine industry.

Product performance was defined by Browning as the technical performance, cost to customer, and delivery timeframe. According to the IPD process defined in section 3.2 technical performance requirements are set at the beginning of the program in the concept definition phase and are listed at the beginning of the requirement documentation. The Memorandum of Understanding (MOU) signed with the airframe manufacturer or government customer outline the promise by the company to deliver on the technical requirements. Failure to deliver on the technical performance results in penalties that are paid to the customer or in the case of the military programs possible cancellation of the program. In addition to the technical requirements a level 1 schedule is included indicating the promises to the customer as far as the delivery timeframes are concerned. Deliveries are defined for engines that are required to support flight testing of aircraft. Flight qualifications, which require a validated design, are required prior to flight testing of the aircraft. Initial production engine deliveries for the launch customer are also defined within the documentation. The cost to the customer is negotiated with the military program office or the various airlines. Since cost to the customer for the developed engine is promised at the beginning of the program, the business case for the product is at risk if the development cost is higher than planned. So, performance uncertainty relative to the value to the customer is highly defined at the beginning of a program. If the product doesn't meet the technical requirements cost and schedule will be affected as the project continues until the technical performance is met or negotiation lead to agreements on penalties levied for shortfalls can be concluded. See Figure 4-5 for the Causal Framework of Factors Contributing to Performance Uncertainty.

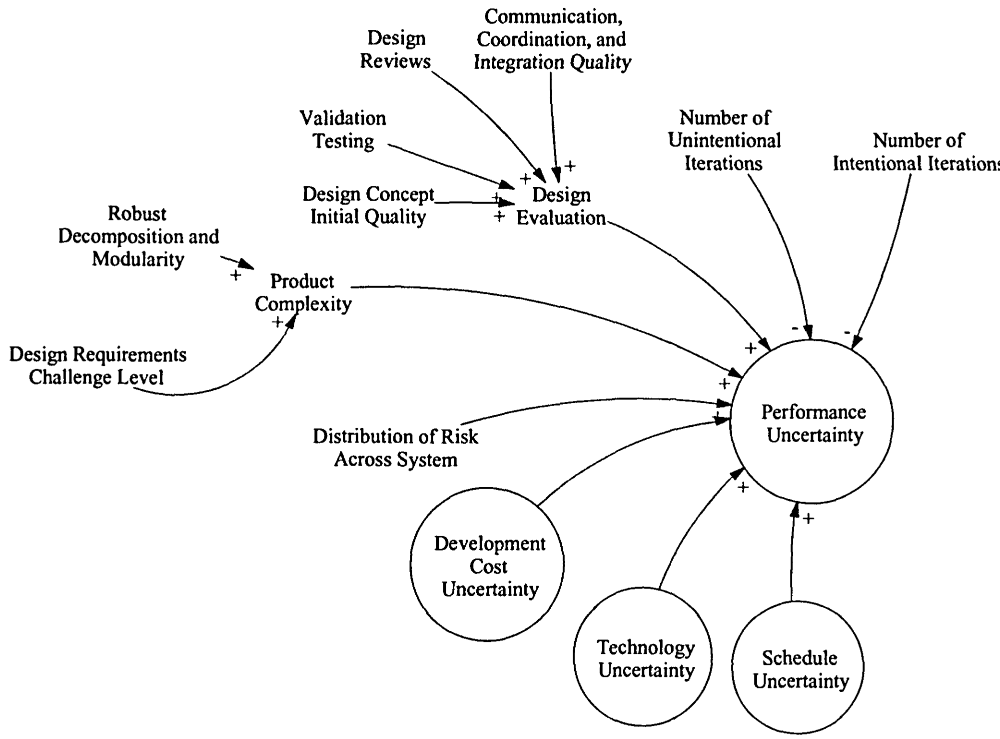

Figure 4-5 : Factors Affecting Performance Risk

The elements of performance risk will be discussed in the following sections.

4.3.1 Design Evaluation

Design evaluation within this framework is the main risk mitigation process. The individual CIPT's and IPT's are responsible for providing designs that meet the individual performance goals as defined by the IPD process. Risk assessments at the program and product level are required at each passport review. In addition there are Module level and Part level reviews at the CIPT and IPT level. These formal reviews are intended to identify problems with meeting performance goals and report on progress. There are also informal reviews that are brought up on a daily basis. The systems engineers may call additional reviews if there are technical issues that are deemed to require more frequent monitoring. The communication, coordination, and integration quality is addressed by the organization and development process and the ability of the company processes and organization to mitigate schedule risk was discussed above.

4.3.2 Design Concept Initial Quality

The company recognizes that the amount of work that is required to produce an acceptable design that meets all customer requirements is dependant on the on the quality of the design concept. Through the ACE process teams have been formed to evaluate the key drivers of risk to first design quality. The team will define changes in the processes that need to be made to allow for a better starting point for the preliminary design which will allow for less iterations and lower performance uncertainty. From the perspective of the Browning framework the presence of the ACE team mitigates the risk and the key drivers identified by Browning will not need to be evaluated.

4.3.3 Verification and Validation Testing

As defined in Section 3.3 the systems organization at Pratt & Whitney that is responsible for system level verification and validation testing is the System Engineering - Validation group. The group specifically plans for the testing that is responsible for achieving regulatory approval for the engines to be able to fly. The Performance Systems Analysis systems engineering group is responsible for using simulations and models to help verify and validate system performance. The group is also responsible for analyzing the data obtained during system level testing. The CIPT's and IPT's are responsible for component and part verification and validation testing respectively. Organizationally, roles and responsibilities are clearly defined for verification and validation of the engine. The various groups are responsible for managing testing within cost and schedule constraints. Therefore there is a very rigid and defined process for managing the amount of verification and validation testing.

4.3.4 Product Complexity

Product complexity of the gas turbine engine has been shown to be extremely high. Drivers of complexity do not need to be discussed as the inherent nature of the complexity makes it difficult to mitigate risk. The entire IPD process and the company organizations have evolved to deal with this complexity. The new initiatives have also been instituted to help achieve the required performance of the engine.

4.3.5 Distribution of Risk across the System

Browning's discussion relates to subsystem risk and the amount of risk each subsystem must bear. Many subsystems with lower risk and a few with high risk have a lower performance uncertainty than many subsystems with high risk. The belief by the company that achievement of a design that has demonstrated system performance in a relevant environment prior to detail design is an indication that the IPD process is working toward all systems with low risk.

4.3.6 Summary

Since the performance uncertainty in the gas turbine engine product development process is highly controlled, the evaluation of the effects of the mitigation policies will be limited to cost and schedule effects. The modeling philosophy will be that the product development process, to produce the given performance in the product, must complete a set number of tasks.

The performance requirements are set early in the program and agreed to by the customer. Requirements at a high level are well defined. Further definition of the requirements below the high level is performed as the product development process progresses from concept to detail design. The product is also known to be highly complex. Reviews are scheduled often within the development process to help identify problems in meeting customer requirements. Information technology is being employed in an attempt to identify performance risks faster. There is a realization that subsystems must be evaluated for technical maturity to more fully mitigate performance risks. If technical maturity is at risk the company has a long history of risk mitigation through carrying parallel development efforts for those parts. Parallel efforts add to cost and schedule uncertainties.

4.4 Technology Risk

Technology uncertainty is a subset of performance uncertainty. Technology risk is defined by Browning as the uncertainty about the ability of a technology to provide anticipated performance benefits within the cost and schedule requirements. The gas turbine engine is a mature product. Any increase in performance is incremental. Examples of incremental changes are new airfoil designs driven by better CFD analysis, more efficient combustor designs, and advanced material coatings for the higher durability of the turbines. The changes are focused at arriving at lower TSFC levels, lower weight, or less noise. All new product development programs are dependent on a number of incremental changes in technology to provide the performance required by the customer. The lifecycle of a gas turbine engine can be upwards of 30 years. Acquisition costs of an engine are high. Unless the product is differentiated from existing products in a significant way customers will be unwilling to take a risk on a newly designed engine. Technology uncertainty in the gas turbine engine product development process is a given and all manufacturers are working on ways to mitigate the risk. See Figure 4-6 for the causal framework for factors contributing to technology uncertainty.

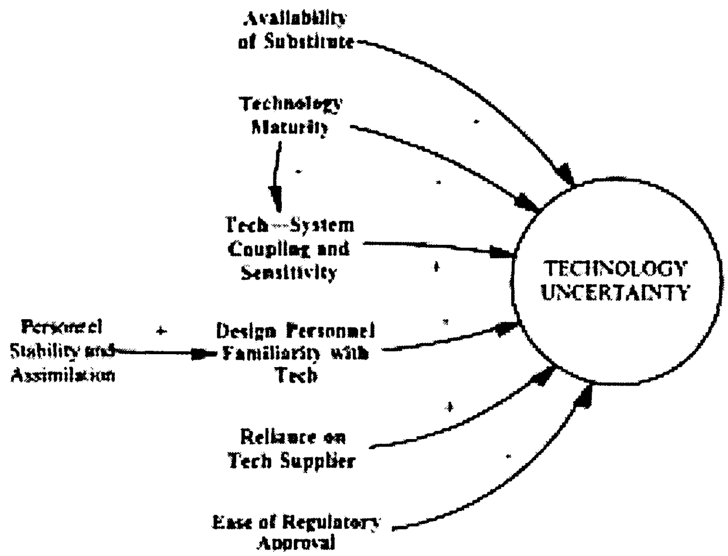

Figure 4-6 : Factors contributing to Technology Uncertainty (Browning)

The most prominent method that is now being employed to mitigate technology risk is the idea of technology readiness level (TRL). The idea of technology readiness levels was driven by NASA's desire to reduce development times in the early 90's. Technology Readiness Levels standardized the approach to assess the technical maturity level of a system or subsystem. The TRL levels are shown in Table 4-2.

Stage	TRL	Objective
Basic Technology	1	Basic principles observed and reported
Feasibility Research	2	Technology concept and/or application formulated
	3	Analytical and experimental critical function and/or characteristic proof of concept
Technology Development	4	Component and/or breadboard validation in a laboratory environment
Technology Demonstration	5	Component and/or breadboard validation in a relevant environment
	6	System or subsystem model or prototype demonstration in a relevant environment (ground or space)
System/subsystem Development	7	System prototype in a space environment
System test, launch, and operations	8	Actual system completed and "flight qualified" through test demonstration (ground or space)
	9	Actual system "flight proven" through successful mission operations

Table 4-2 : NASA Technology Readiness Levels:[27]

Recently GE announced a new gas turbine engine product development process that takes 18 months. The keys to shortening the product development process is the notion that all technologies required for a commercial product must be proven and mature prior to launch.[28] Pratt & Whitney also realizes that technology maturity is key to reducing technology risk and is taking steps to institute processes to ensure that technology readiness levels are met. During the passport review processes new technology must be reported on relative to the ability of the CIPT to deliver a TRL6 component prior to Detail Design. The requirement is noted in section 3.2. The product development process as defined indicates that the TRL level must be reported on relative to the plan to boost the technology level to level 6 prior to detail design. The risk mitigation strategy of substituting technologies as an alternative to required technologies would not be required if a program could be all TRL6 prior to launch. In the past this strategy was the one typically deployed during a program. The problem with substituting technology is the performance goals may not be fully met, the cost is high to carry multiple options, and the time required to implement the substitute technology is not planned. Today substitution risk mitigation strategies are still employed if the technology has not achieved TRL6.

Technology and system coupling is the idea that component implementation in a system will not function as planned due to coupling. Since the gas turbine engine development process recognizes this problem the mitigation is planned for during the technology maturation process.

The idea of regulatory approval is always factored in during early stages of the planning process and is accounted for during product development process. The validation process is geared toward achieving regulatory approval.

---

[27] Mankins, John C. "Technology Readiness Levels: A White Paper". NASA Advanced Concepts Office, (http://www.hq.nasa.gov/office/codeq/trl/trl.pdf). April 6, 1995.

[28] Kandebo, Stanley, "General Electric Aims at 18 Month Engine", Aviation Week and Space Technology (November 2002)

The cost of development for a completely redesigned engine has become prohibitive in today's economy. Almost all programs are instituting partnering. Reliance on a technology supplier is a real risk to a development program. The uncertainty introduced by not having control of the process is great. Pratt and Whitney has realized this fact and has instituted lean practices to mitigate the risk. The suppliers have full CIPT and IPT responsibilities and are co-located with the design teams. Risk mitigation strategies are in place.

The last driver of risk for technology risk is the idea that the design personnel are familiar with the technology. The familiarity is dependent on staffing stability and training. Experienced personnel are the drivers of incremental changes in technology. Experienced staff trains and mentors the novice and intermediate staff. Within the company there is a drive to identify proficiency levels of the personnel in the company. Proficiency is tied to the ability to execute the standard work the particular group is responsible for carrying out. One of the measurement metrics the leadership of the company is watching is the amount of personnel at each level of proficiency. Training used extensively to allow staff to move up the proficiency ladder faster. There is a great deal of in-house classes that the development staff is required to take to become proficient in their jobs. Key personnel are identified in the process and the hope is that the groups will do everything possible to retain these people. Staffing stability can be negatively affected in the face of instabilities in the workload. Retention of full staffing levels through downturns related to the instability described at the outset of the thesis is difficult as budgets for the programs are reduced in the valley of the cycles.

4.5 Market Uncertainty

Market uncertainty is defined by Browning as the uncertainty in the anticipated utility or value to the market of the chosen "design to" specification. In the case of a gas turbine engine a program is not kicked off unless the customer has signed contracts for what is deemed an adequate number of engines to make program launch feasible. The program is supposed to be cancelled if according to planned numbers there is no business case. The long lifecycle of the engine makes determining the market difficult. An engine may not see an expansion of the market until years after program launch. The expansion may not have been anticipated. The reverse case is also true. The Memorandum of Understanding outlines delivery dates for initial production engines for the launch customer. For these reasons market uncertainty in the gas turbine engine business is not a big driver in the overall uncertainty of the product development process. The market will most likely not respond differently than the business projected prior to the completion of the development process

4.6 Business Uncertainty

Business Risks according to the definition in Table 4-1 : Categories of Product Development Risk (Browning, Tyson) are political, economic, labor, societal, or other factors. The scope of this thesis is directed at reduction of risks that are controllable by the internal processes of the company. Business uncertainty is not within the scope of this thesis and will not be addressed.

4.7 Conclusion

Cost and schedule risks are the main factors that must be reduced within the processes. The other risks within the gas turbine development process all serve to increase cost and schedule risks. The framework for analyzing risk within programs indicates that programs are planned with highly aggressive schedules. The assumed productivity of the staff is assumed to be very high. Company wide initiatives to define processes are aimed at improving the productivity of the workers. Since productivity levels are key to reducing the cost and schedule risks, the improvement in productivity will be evaluated for the ability to overcome the problems that are normally associated with the gas turbine product development process.

The assumed quality of the work is assumed to be high. An overall corporate philosophy of focusing on quality implements the ACE process to continually improve quality in all processes including the product development process. Special tiger teams have been formed to focus on specific issues that affect cost and schedule risk as well as initial quality of the design. The product development process outlines a review process to identify risk and implement risk management plans to reduce the amount of rework that is required in the programs. An Earned Value Management System has been employed to help identify process problems earlier so that they may be addressed faster. Since quality is an overarching principle at the company the thesis will evaluate the ability of quality improvements to overcome the problems that are typically present in the gas turbine product development process.

The issue that analysis of the risk framework identifies as an issue that is not well addressed is resource availability. The drive to minimize cost on all aerospace product development programs forces available budget to be minimized. Staffing is a main component of program budgets and is therefore minimized on programs. The minimization of staff is causes more schedule uncertainty because of the instability of the workload inherent in the company. The motivation section of the paper outlines the current situation of the product development programs at the company. Figure 1-1 shows the instability of the workload as the product development process progresses from the completion of one program to the initiation and execution of three programs. The resulting staffing plan is shown in Figure 4-7. The drive to minimize the staff coupled with the instability of the manpower requirements causes difficulty in maintaining adequate resources to complete programs. The reluctance to engage in a hiring and firing cycle of permanent workers due to availability of budget makes understaffing during peak workloads commonplace. Since staffing is identified in this section as a major driver to cost and schedule risk the thesis will evaluate ways to enhance staffing within the programs. The methods identified are overtime use, outsource labor use, and alleviation of bottlenecks by lower utilization of resources. The system dynamic model will evaluate the effectiveness of the use of these policies in reducing the cost and schedule growth within the multi-project environment present within the company.

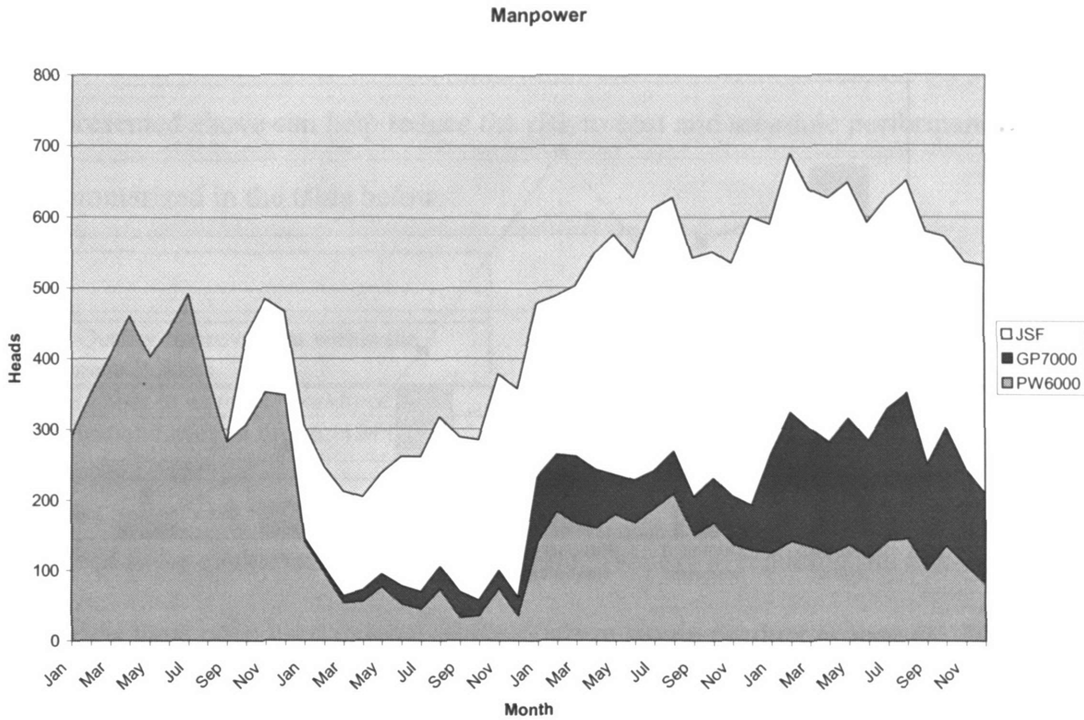

Figure 4-7 : Staffing Plan for Completion and Execution of Current Programs

Data collected from past programs indicate that the issues outlined here are in fact main drivers of the risk in programs. Manpower has a high number of citations as a reason for schedule variance as well as being the largest driver of schedule variance. Out of Plan work or unplanned iterations are the main driver of cost variances. See Figure 4-8 for the Pareto of variances. The thesis will use these conclusions to define policies to mitigate cost and schedule risk to be evaluated using a multi-project systems dynamics model in the following sections.

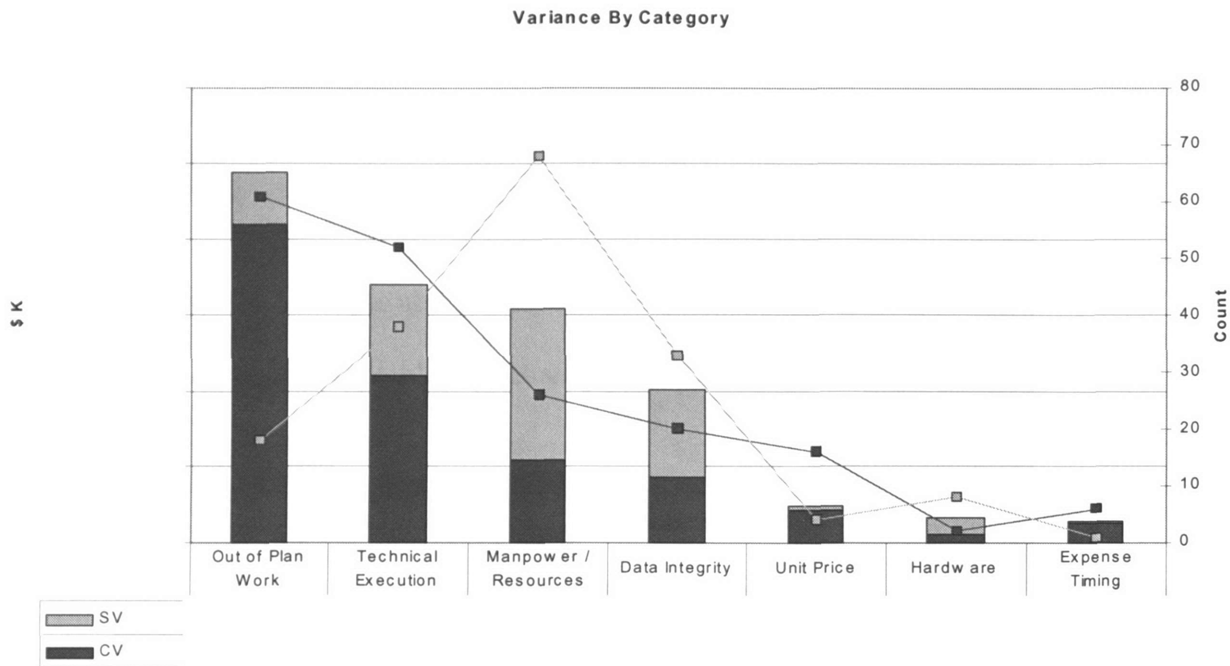

Figure 4-8 : Pareto of Variances on Past Programs

5 Systems Dynamics and the Multi-Project Staffing Model

The policies presented above can help reduce the risk to cost and schedule performance.. These policies are summarized in the table below.

Policy
Use of Overtime
Productivity and Quality Improvement within the Product Development Process
Use of Outsource Labor to increase Workforce
Staff at Low Utilization Levels at the onset of the Product Development Process

Table 5-1: Policies to Decrease Risk to Cost and Schedule

The policies need to be evaluated relative to their effectiveness in reducing the risks. System Dynamics models have been used extensively to analyze single product processes. Many such models were found during the literature review. The high workload due to three concurrent programs following a period of low workload and the resulting staffing profiles presented in this thesis requires a multiple project viewpoint to adequately model the system dynamics. The risk analysis points to the methods to enhance the availability of staffing and their ability to perform at high productivity and quality levels given the instability in workload. A Systems Design and Management thesis by Karl Pilon and Greg Herweg developed a multi-project model that permits the exploration of manpower resource allocation decisions. The model envelops both the structure and the processes that represent technology product development at Sikorsky Aircraft and Xerox Corporation.[29] This section will relate the current structure and processes at Pratt & Whitney as presented in Section 3 to the processes and structure made by the authors of the systems dynamic model to show the ability of this model to analyze the policies related to staffing, productivity, and quality. Also, the value of intellectual capital and the dynamic hypothesis underlying the model and the applicability to Pratt & Whitney's situation will be discussed.

[29] Herweg, Greg and Pilon, Karl, System Dynamic Modeling for the exploration of Manpower Project Staffing Decisions in the Context of a Multi-Project Enterprise. SDM Thesis, (Feb 2001)

5.1 Workflow

The Herweg-Pilon model uses the flow of work in the enterprise and the relation to completion of projects. The model is designed to allow for the simulation of work completion, both correctly and incorrectly, and it's movement through the four phases of a representative process. This section will show that the four phases that are represented in the Herweg-Pilon system dynamic model are representative of the phases of the Pratt & Whitney IPD process as defined in the background section of this paper without modification. In the context of this thesis the Intentional Iterations and Unintentional Iterations are modeled through phases of the product development process. Note that the authors of the system dynamic model assumed four phases in the product development process. Those phases are the Requirements Phase, High Level Design Phase, Development Phase, and the Test Phase. The phases of the program were taken from an adaptation of a Quality Function Deployment framework developed by Slack in a 1998 Masters Thesis entitled "The Application of

Lean Principles to the Military Aerospace Product Development Process".

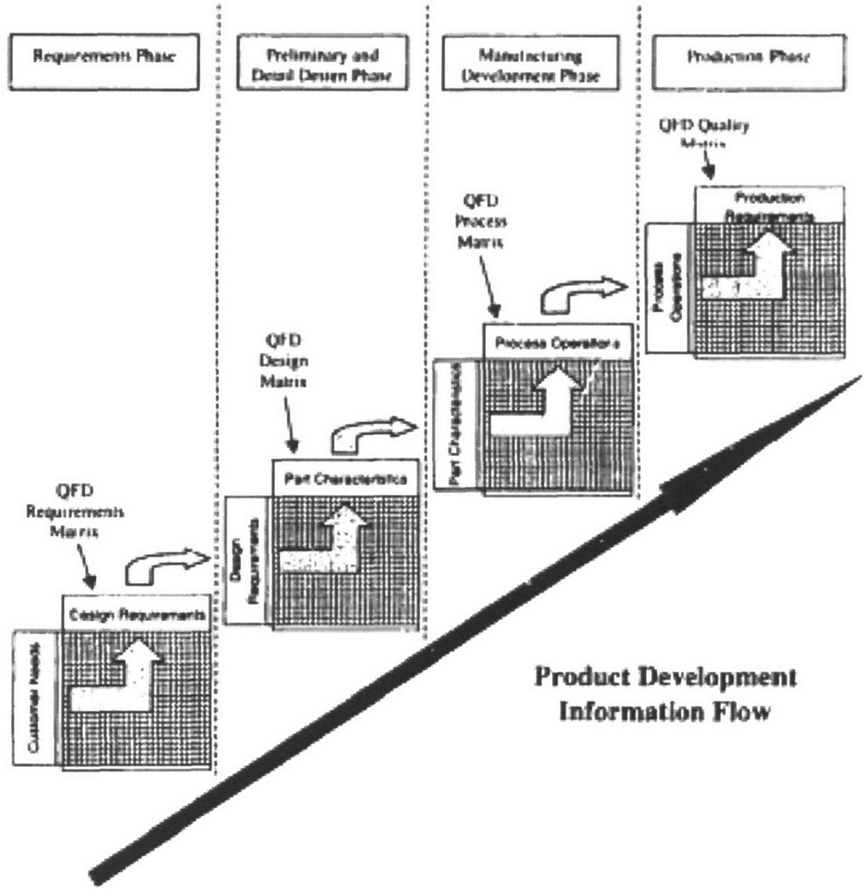

Figure 5-1 : Phases of Product Development Process; Slack

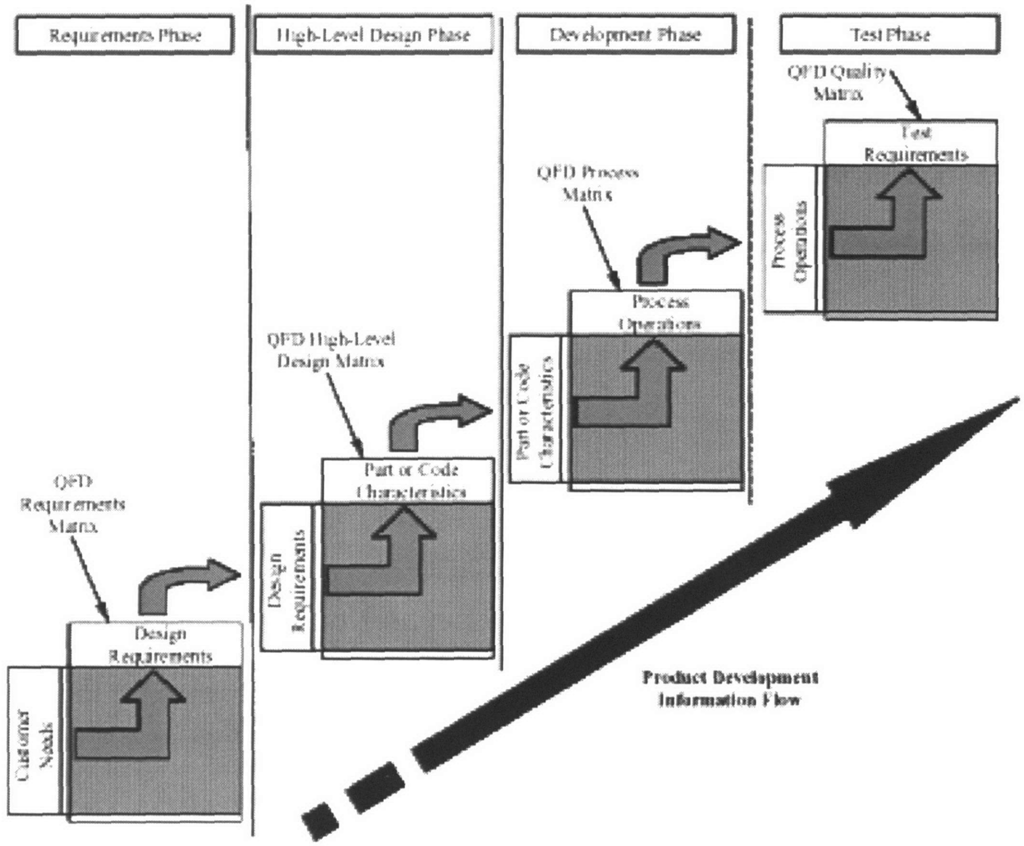

Figure 5-2:Phases of Product Development Process; Herweg and Pilon

The framework aids in visualizing the value creating activities of transforming customer requirements into design requirements. The design requirements can than be transformed into part and component characteristics that must be developed and finally the parts and components can be validated. The QFD framework proposed by Slack and adapted to the systems dynamic model is one of what could be many frameworks, but the underlying idea of all of the frameworks he presented is that they represent the "what" domain. In other words, the framework shows what must be accomplished in the product development process. This representation does not address how to accomplish these development tasks. Since the model employs a framework that merely maps to what must be accomplished it becomes easy to map the six phases of Pratt & Whitney's product development process to that of the Herweg-Pilon model. Pratt & Whitney's Concept Initiation Phase and Concept Optimization Phases map to the Requirements Definition Phase based on the background discussions. The Preliminary Design Phase maps to the High Level Design Phase. To more easily break up the Pratt & Whitney product development process as currently envisioned the third phase would entail mapping the Detail Design Phase to the Development Phase. Finally, the Verification and Validation Phases map to the Test Phase. In this manner the four phases that represent workflow in the Herweg-Pilon model can be adapted to the Pratt & Whitney IPD process.

5.2 Organizational Structure

This section will show the system dynamic model developed by Herweg and Pilon is applicable to Pratt & Whitney since the organizational structure the model is base upon is similar to Pratt's. The Herweg-Pilon model used Sikorsky Aircraft and Xerox Corporation as the basis for the model. Sikorsky Aircraft is also a division of United Technologies Corporation and has had a similar history relative to the organizational evolution as that of Pratt & Whitney. As explained above Pratt & Whitney has evolved from a functional organization to a hybrid matrix organization utilizing heavyweight project teams. Functional managers and heavyweight project managers guide integrated Product Teams. The systems organizations are tasked with integrating the components since the gas turbine engine has a high degree of coupling. During the 90's "Sikorsky committed to a change intended to realign division resources in order to maximize value to the customer and improve competitive advantage. This change, consisting of a change from a functional organization to that of a platform organization, was initiated within the engineering department in February 1998"[30]. This change created autonomous product platform teams that would be collocated. The teams would be comprised of individuals from all of the functional branches, which would be responsible for all aspects of the aircraft development process. Sikorsky's structure still retained the functional core competencies although in a much smaller capacity. The consolidation of the core competencies

[30] Herweg, Greg and Pilon, Karl., "Systems Dynamics Modeling for the Exploration of Manpower Project Staffing Decisions in the Context of a Multi-Project Enterprise", Masters Thesis SDM Program, MIT, Jan 19, 2001, pp 27-31.

allowed the company to reduce resources in a similar manner as Pratt & Whitney. The core competencies were deployed to the product platform teams when each new product was launched. Sikorsky is much later than Pratt & Whitney in the evolution from a functional organization to a matrix organization. They may not have driven to a heavyweight structure as in the case of Pratt & Whitney, but the organizational structure in both companies is basically a hybrid between a functional organization and a pure project organization. The Herweg-Pilon model depended on the interdependencies when the hybrid organizations are created. The move toward a smaller functional organization erodes the functional expertise. The competition for scarce resources is the basis for the multi-project aspect of system dynamic study. From the perspective of the organizational structure the Herweg-Pilon model is directly applicable to the situation at Pratt & Whitney.

5.3 Intellectual Capital

The system dynamic model developed by Herweg and Pilon is also based on the growth or shrinkage of intellectual capital with the multi-project product development process. The discussions relative to risk mitigation in this thesis have centered on staffing as being the weakness of current product development system at Pratt & Whitney. The question that needs to be answered becomes what is the real issue that a lack of staffing or the erosion of functional expertise drives. The idea of intellectual capital is the key issue. The definition of intellectual capital has been a difficult one to answer, but Thomas A. Stewart settled on one in his book Intellectual Capital: The New Wealth of organizations. He settled on Professor David Klein and consultant Laurence Prusak's definition as presented:

"Intelligence becomes an asset when some useful order is created out of free-floating brainpower - that is, when it is given coherent form (a mailing list, a database, an agenda for a meeting, a description of a process); when it is captured in a way that allows it to be described, shared, and exploited; and when it can be deployed to do something that could not be done if it remained scattered around like so many coins in a gutter. Intellectual capital is packaged useful knowledge."[31] At Pratt & Whitney the current attempt is to create Standard Work. Standard Work is an effort to "package useful knowledge". Experts who know how the process for developing a gas turbine engine works need to package that knowledge so that those who are not experts can use it. Stewart defines this knowledge capture as the semi-permanent body of knowledge. The use of standard work will improve the ability to transfer the knowledge. The transfer of this tacit knowledge from experts to intermediate to novice skill level employees is the main way intellectual capital can be increased within the company. By increasing the intellectual capital within the existing workforce productivity can be increased. The model will test the ability of increasing intellectual capital, and therefore productivity, at a faster rate at reducing the risk to cost and schedule posed by problems in the product development process. This is one method that increases in productivity will be tested in the model.

5.4 Dynamic Hypothesis

The Herweg-Pilon Model is based on the following dynamic hypotheses.

Attrition

Productivity

Product On-Time Delivery

Resource Progression from Novice to Intermediate to Expert Skill Level

Timing of Hiring

This section will relate the product development system at Pratt & Whitney to these hypotheses to prove the validity of the model in being able to determine the system dynamic outcome of the tested policies.

[31] Stewart, Thomas A., "Intellectual Capital: The New Wealth of Organizations", Doubleday, 1997, pp66-68.

5.4.1 Attrition

The first dynamic hypothesis that the Herweg-Pilon Model described is the idea that high workloads drive attrition within the company. There are three loops related to this hypothesis. When the work to do is high the workforce required to accomplish the tasks on time is increased. The added workforce will produce more errors and rework. The management response to the work levels rising is to increase overtime. Overtime raises the amount of work done and reduces the work that remains. There is also a rework loop where the added overtime increases fatigue which in turn will decrease the quality of work. The last loop shows where attrition is driven. The high levels of fatigue also drive higher attrition. The higher attrition rates cause a decrease the workforce. The causal loop structure is shown below.

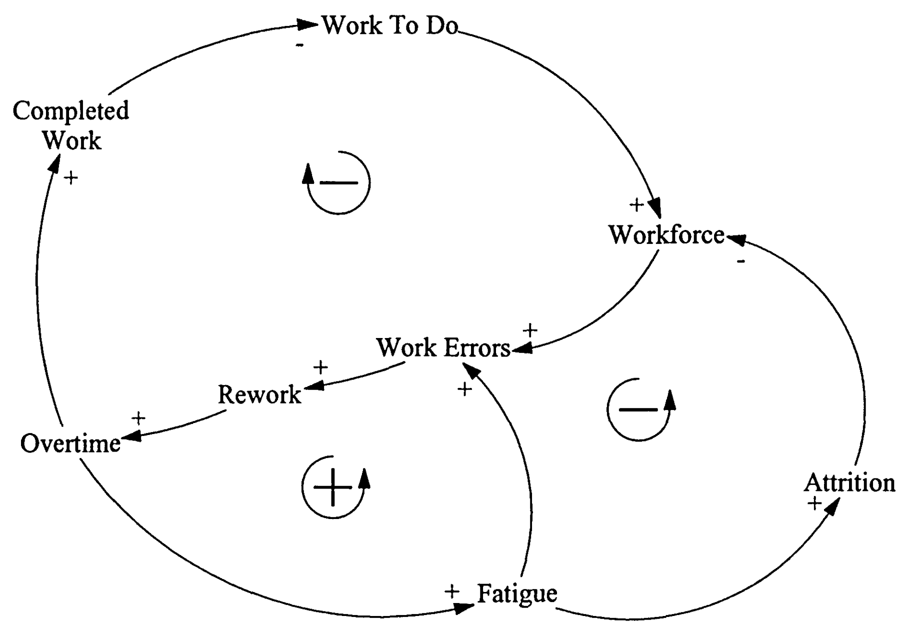

Figure 5-3 : Causal Loop Diagram for Attrition

As a result of the move from Florida to Connecticut many people in the validation group were lost due to attrition. Approximately 20% of the staff left the programs. The remaining staff could not be replaced and those employees left were had to endure constant overtime. The constant long hours driven by high overtime were taxing to the young employees. The sense of accomplishment was outweighed by the fatigue. By the time the programs had reached the later phases of the program the work to do to meet the schedules had risen to high levels. As the work to do rose workforce was added. The addition of workforce included people who weren't as experienced with the process the number of errors rose. This effect was more pronounced on the commercial side where staffing was lower during the life of the program. The overtime did keep the programs from falling further behind so more work was getting done. The effects of fatigue and lower quality caused less work than planned to be accomplished with the use of overtime in the programs. The attrition loop is difficult to understand, as the market place for jobs can be the ultimate determinant for the amount of attrition. In the case of Pratt & Whitney attrition was high when the ability to move jobs was easy and the late 90's boom caused starting salaries to soar for people moving jobs. In today's climate the attrition is much lower even though overtime levels are still high. The main loops which are in effect at Pratt & Whitney have more to do with the work that required to be completed and the rework that is driven due to fatigue and lower quality.

5.4.2 Productivity

The second dynamic hypothesis put forth by Herweg and Pilon is that employees overburdened by high workloads and too many different projects suffer a loss of productivity. They further believe that the loss of productivity drives a slowdown of skill advancement and learning. The causal loop is shown below.

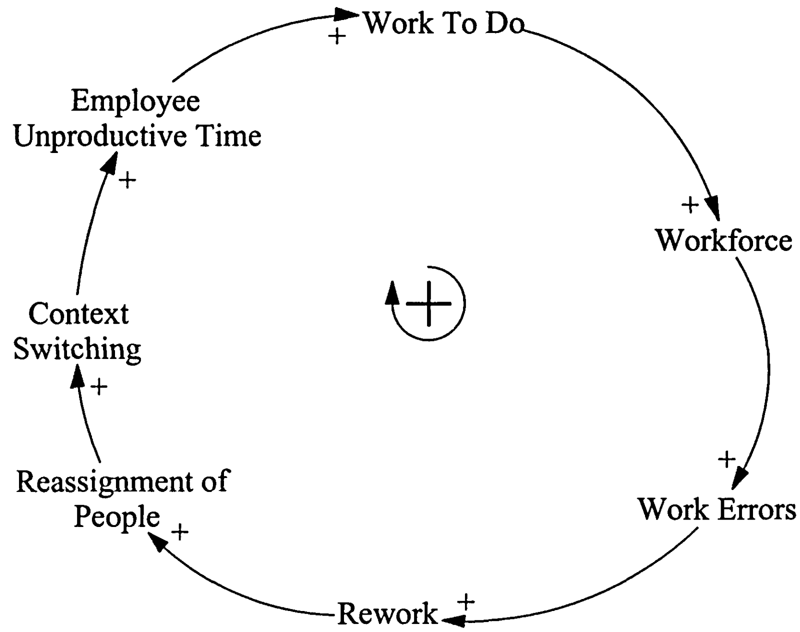

Figure 5-4 : Causal Loop Diagram for Productivity

At Pratt & Whitney the common situation that employees find themselves in is in a high workload condition. The drive to keep costs low causes a lack of budget and less than staffing levels. The quality of the work suffers and mistakes are made. A measure of how overburdened the workforce can be is the data related to turnbacks. Turnbacks are the documentation of work errors and over the course of a year only half of the tasks are addressed. Under this situation the workforce finds tasks that they are most efficient doing and works on those tasks. Learning new tasks is only accomplished when people leave. Tasks that are required to be done can only be completed when others learn new tasks. Reassignments also occur when tasks are left undone under the high workload and high priority projects pull staff to complete the projects.

5.4.3 Project On-Time Completion

The third dynamic hypothesis that the Herweg-Pilon system dynamic model is based on is related to Project On-Time Completion. The completion of projects is delayed when due to the discovery of rework in work that has been completed. Herweg and Pilon believed that 3 additional causal loops were involved in the system dynamic of slowing the project progress other than those described above that are related to completion of work to do and attrition. Three of the causal loops were represented by work errors.

The first of the three loops is a reinforcing loop where work errors result in more rework and more overtime. The additional overtime increases employee fatigue and drives attrition. Employees leaving the workforce reduce intellectual capital and means less skill available. The lower ability of the workforce causes more work errors.

The second reinforcing loop postulates that project schedule slip is the result of the increased amount of errors. As schedule slip increases, project on-time completion decreases. Less projects that are completed, which serves to decrease learning and intellectual capital. A decrease in skills produces an organization that is more error prone.

The third loop associated with project on time completion is driven by work errors. Work errors cause rework and overtime, which can drive more completed work. Competed work drives improved learning. Improved learning increases intellectual capital. The loop is therefore a balancing loop.

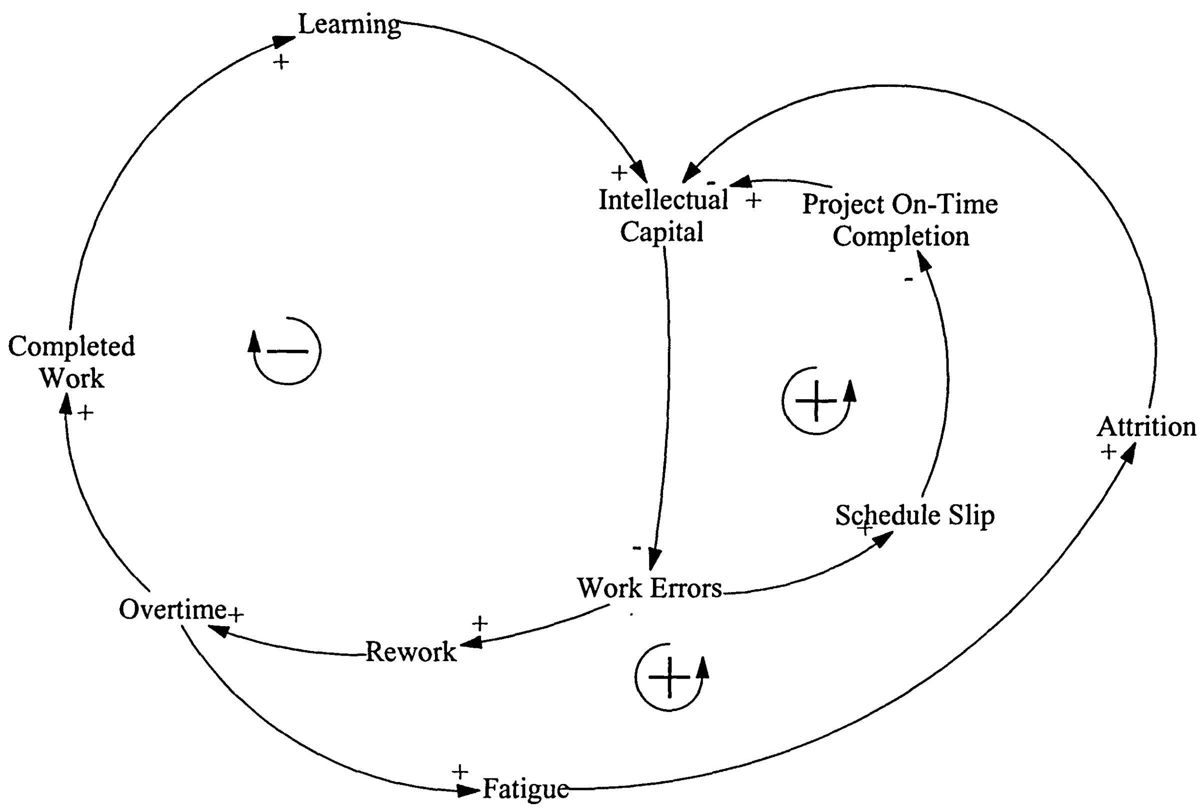

Figure 5-5 : Causal Loop Diagram for Project On-Time Completion

At the company the first loop is in evidence, but reorganizations of the company have clouded the issue. The move from Florida to Connecticut has resulted in a great deal of attrition. The loss of people has reduced the intellectual capital. Rework, overtime, and the amount of work is increased for the staff that remains. Learning is increased, as the staff has to take on new tasks. The growing ability of the workforce to perform the tasks increases the company's ability to complete projects. The more projects that the staff can complete the more knowledge they will accumulate. The last loop negatively affects the ability of Pratt's organization to complete the project on time although the business climate lessens the impact. The accumulated overtime and fatigue can cause additional attrition. The attrition will reduce the intellectual capital. Since the market place for hiring has been slowed the attrition at the company is not as even as recently as a year and a half ago.

5.4.4 Resource Progression from Novice to Intermediate to Expert Skill Level

Resource progression is the basis for the fourth dynamic hypothesis that the system dynamics model is based on. In this causal loop intellectual capital growth is through employee learning. Mentoring is one method to transfer knowledge from more skilled employees to less skilled employees. Mentoring is more difficult if the rework drives the more skilled employees to be moved onto the high priority jobs to keep projects moving.

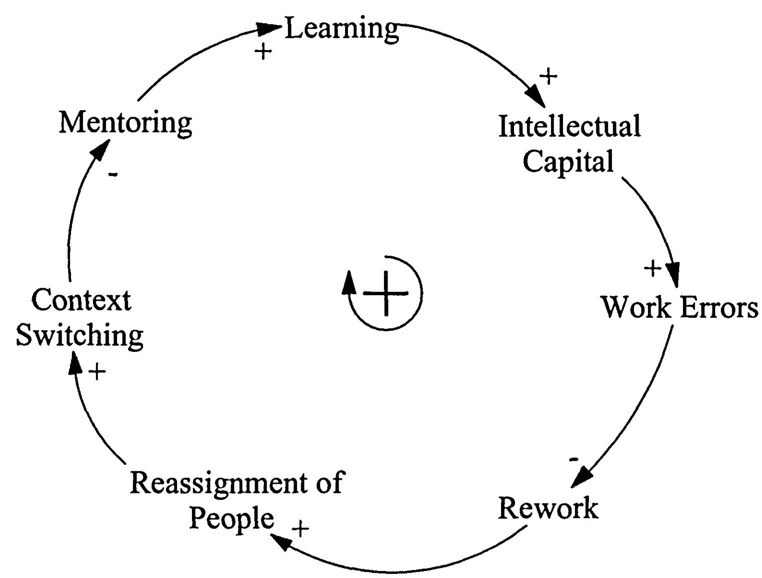

Figure 5-6 : Causal Loop Diagram for Resource Progression

The current situation at Pratt & Whitney shows evidence of this causal loop structure. The workforce is smaller and the intellectual capital is lower than the past. More work errors are evident and because projects need to be kept on schedule higher skilled personnel are moved to critical path jobs. The flexibility of the programs allow for some ability to mitigate the schedule risk, but certain problems usually get the attention of the most skilled practitioners. The movement of the higher skilled employees leads to less mentoring. There are many situations at the company that no one was trained for and the skilled employees who knew how to do the work retired. The retired employees are usually rehired as contract labor to continue to work high priority jobs.

5.4.5 Timing of Hiring

The last hypothesis that Herweg and Pilon based their system dynamic model on addressed the timing of hiring. Hiring of employees is initiated when staffing levels fall below the perceived ability of the current workforce to handle the workload. Overtime is at the highest levels possible. When hiring is initiated the average skill level of the workforce will drop and there is more of a requirement for mentoring with the added new hires. Learning and intellectual capital can be increased with more mentoring. Quality of work is increased so rework, overtime, and fatigue are reduced. The hope is that the hiring will be stopped as skills go up.

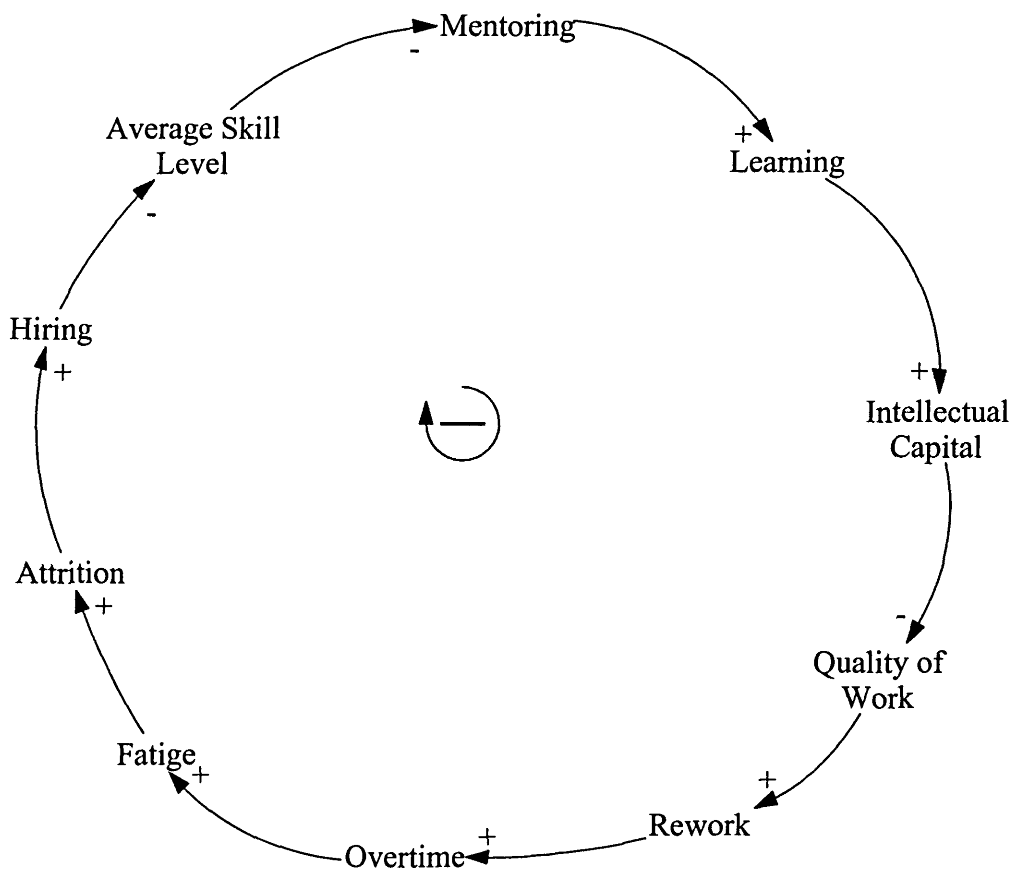

Figure 5-7 : Causal Loop Diagram for Hiring

This particular loop at Pratt & Whitney is very weak. As the risk framework showed the resources are driven by available budget. Hiring must first be able to be justified by the budgets. Once the budget justifies the need for additional manpower the company searches for available people within the company. If that process fails the hiring process can begin. The hiring process is slow and cumbersome, so by the time a need is initially identified the hiring process can easily take 8 months to a year. The link between attrition driving a need for hiring is very weak.

5.4.6 Summary

The above sections indicate that the dynamic hypotheses that the Herweg-Pilon model is based on are in evidence at the company and representative of the system. Because the dynamic hypotheses are representative of the system the base structure of the system dynamic model can be utilized to understand the dynamics of multiple projects performed in the product development process at Pratt & Whitney. The base structure will be used without alteration. The base structure is defined as the work to do structure that is arranged in a four-phase workflow. Within the base work to do structure there is a section dealing with the relative attractiveness of the projects, which moves staffing from one project to another. There is additional structure in the model that is labeled as the Effects portion. This section calculates how things such as skill level effect quality and productivity. Quality and productivity have been identified as issues that need to be studied for the ability to effect the outcome of the multi-project product development process. Also, there is a people section of the model that computes the productivity expected at any time based upon the number of workers, their skill level, and their level of fatigue. This section of the model will be modified as outlined in the following sections to evaluate the policies that relate to productivity, staffing, and skill level. The base structure of the model is able to simulate the planned multi-project product development process at Pratt & Whitney with the work that must be accomplished and the related staffing levels. The constants that are contained within the structure that is present can be changed to test the policies related to staffing, productivity, and quality. Some structural additions will have to be made to the model. The changes that are made to the model will be described in the following section. For a more detailed description of the structure of the Herweg-Pilon system dynamic model refer to their thesis.[32]

The causes of hiring and attrition are weak and affected by other causal loops that are not modeled. Hiring is a long drawn out process that is not effective in adding workforce quickly. Attrition has been reduced due to the recession. The ability to move from company to company has been reduced over the last year and half. The decline in the aerospace and airline industry and the effect of Enron has severely limited the options of gas turbine engineers. The largest amount of hiring of gas turbine engineering talent was the power generation market. Enron reduced the venture capital available for market expansion. The loops reflecting loss of productivity, the amount of work to do, the factors affecting project on time completion, and the learning ability of the organization are much more in evidence. The causal loops related to work errors and productivity are very strong and effect the system dynamics to a higher extent. They are internal to the organizations and are highly affected by overtime use. Overtime is used to a high extent. Engineering organizations were put on mandatory overtime for periods as long a 9 months. Examples can be found through the company were overtime was still worked even when tasks were ahead of schedule because of the fear that rework will inevitably force the program behind schedule. The basic ability of the Herweg-Pilon system dynamic model to represent Pratt & Whitney's work structure and organization is in evidence.

---

[32] Herweg, Greg and Pilon, Karl., "Systems Dynamics Modeling for the Exploration of Manpower Project Staffing Decisions in the Context of a Multi-Project Enterprise", Masters Thesis SDM Program, MIT, Jan 19, 2001

6 Analysis of Staffing Policies

Based on the above analysis of the weakness in the current product development process at Pratt & Whitney the systems dynamic model was exercised through seven scenarios. The analysis consisted of calibrating the model with data from the current situation at the company and analyzing the effectiveness of policies that can mitigate cost and schedule risk. The policy recommendations come from evaluating the company with respect to the risk framework described in section 4 and are outlined in Table 3-1. These policies are intended to mitigate the risks outlined in section 4.7. The resulting rework from lower than planned productivity and quality and the inadequate staffing resources were cited as the main drivers of cost and schedule uncertainty. The policies tested are summarized in the table below.

Policy	Discussion
Use of Overtime Scenarios A, B and C	• Typical management response to alleviate schedule risk • Easily implemented at low cost to company
Productivity and Quality Improvement Scenario D and E	• Schedules and resources are planned based on Quality Initiatives and Process Improvements
Use of Outsource Labor Scenario F	• Management tool to handle temporary spikes in workload • Faster response time to hire and fire
Staff at Levels that Queuing Theory Indicate Enhanced Flexibility Scenario G	• Current projects are staffed at 100% utilization for perfect plan • Hire rate creates lag to bring on additional staffing • Queuing theory indicates at above 70% resource utilization rates waiting times are significantly increased • Management resists policy due to higher planned cost

Table 6-1 : Potential Policy Recommendations for Mitigation of Cost and Schedule Risk

Each policy is followed by a scenario designation. The designation will be consistent and map to the analysis that follows performed using the multi-project System Dynamics model developed by Greg Herweg and Karl Pilon and modified in this thesis. The System Dynamics model greatly aids in gaining an understanding of the system and the interactions of these new policies with the current system performance. The model was designed to handle four separate product development programs and four corresponding product development phases in each program. Validity of the model for representing the company structure and processes was discussed in section 5. The following sections will outline the policies and the changes made to the model that correlate the model to the recent historical performance at Pratt & Whitney coupled with the three programs that are currently in works. Detail discussions of the model baseline, effects of the addition of imperfections to quality and productivity, and effects of project unintentional iterations introduced into the projects follow this section. Also, discussions of the effect of current mitigation policies as well as development of the proposed policies are included in the following sections.

6.1 Model Changes

The following section outlines the changes that were made to the Herweg-Pilon system dynamics model to correlate the model to the instability that is present at the company. In addition the model changes represent productivity and quality problems that have been seen during recent programs, but not planned for. Other changes outlined are changes that test the possible policies, which can mitigate the cost and schedule risk to the program and improve the ability of the program to deliver value to the customer. Table 6-2 represents the cases that were evaluated to assess the system performance and the effectiveness of the policy recommendations.

Discussion	Changes to Model
Scenario A Baseline Project - One project finishing, Three parallel projects starting.	• Staff carried over from previous programs • Perfect productivity from all levels of experience • Low complexity • Perfect quality • Average staffing utilization 100% • Low Attrition
Scenario B Baseline Project with Imperfect Quality and Low Productivity	• Productivity levels different for different levels of experience • High Project Complexity • Imperfect quality
Scenario C Baseline Project with Imperfections and Project Instability	• Added effect of unintentional iterations which occur late in the program and cause significant added rework
Scenario D	• Process improvement effect on productivity

Baseline Project (Imperfections and Instability included) with Productivity Improvements	• Training • Standard Work • Shorten time to advance from Novice-Intermediate-Expert • Test Sensitivity
Scenario E Baseline Project (Imperfections and Instability included) with Quality Improvements	• Quality Improvements • Training • Process improvements on reducing turnbacks • Test Sensitivity
Scenario F Baseline Project (Imperfections and Instability included) utilizing Outsourcing	• New Hire quality and productivity equal to intermediate or expert level employees
Scenario G Baseline Project (Imperfections and Instability included) with lower initial utilization	• Begin Project with workforce not at full utilization • No productivity or quality improvements

Table 6-2 : Summary of System Dynamic Cases

6.1.1 Scenario A - Baseline

The architecture of the Herweg-Pilon system dynamics model matched the organizational structures and the base work structures at Pratt & Whitney as described in section 3. The model was recalibrated to better represent the current issues at Pratt & Whitney so that the policies that could help the risks to cost and schedule could be tested. Since staffing and workload instability is central to the problems presented in this thesis the model was calibrated to represent the profiles presented in Figure 1-1 and Figure 4-7. Work To Do was minimized for Project 1 to represent the project finishing. The amount of work to do over the four phases of a project was kept low during the concept and preliminary design phases. The Systems Engineering - Validation group does not have large requirements for manpower during these phases relative to the validation phase. During detail design work to do begins to grow as long lead test systems need to be constructed. Finally, the work to do ramps up immensely during validation testing. Normal Productivity levels for employees were determined from analysis of jobs that are typically open and being worked on a daily basis. The total number of tasks that are required to be worked were chosen to produce the planned level 1 schedule with the perfect productivity and quality and no major problems in the projects. The workload inputs are defined as follows:

	Concept Phase	Preliminary Design Phase	Detail Design Phase	Validation Phase
Normal Productivity	10	10	10	7

Table 6-3 : Productivity Inputs to System Dynamic Model

Project	Work To Do - Concept Phase	Work To Do -Preliminary Design Phase	Work To Do -Detail Design Phase	Work To Do - Validation Phase
1	10	10	10	600
2	1600	5600	11700	50400
3	1000	3500	7540	42000
4	2000	6000	11800	47880

Table 6-4 : Workload Inputs to System Dynamic Model

To accurately model the current situation at Pratt & Whitney the projects were populated with a baseline staff that is representative of what existed at the completion of the two previous programs. The model was modified to allow this staff to rapidly move off the old programs as the new programs began to ramp up. The minimum staffing for each project at the beginning is representative of the initial formation of an organization to look at concepts. Below is the definition of the initial staffing levels for the model.

Project	Concept Novices	Concept Intermediates	Concept Experts	Prelim Des Novices	Prelim Des Intermediates	Prelim Des Experts
Project 1	0.00	50.00	19.00	0.00	50.00	31.00
Project 2	0.00	1.00	1.00	2.00	1.00	1.00
Project 3	0.00	1.00	1.00	1.00	1.00	1.00
Project 4	0.00	1.00	1.00	2.00	1.00	1.00
	Detail Des Novices	Detail Des Intermediates	Detail Des Experts	Validation Novices	Validation Intermediates	Validation Experts
Project 1	0.00	50.00	41.00	0.00	125.00	140.00
Project 2	2.00	1.00	1.00	3.00	1.00	1.00
Project 3	2.00	1.00	1.00	3.00	1.00	1.00
Project 4	2.00	1.00	1.00	3.00	1.00	1.00

Table 6-5: Staffing Inputs to System Dynamic Model

A new variable was added to separate the time to move out of Project 1 so that staff was free to go to other programs at the completion of the programs. The resulting staffing profile is as shown in the figure below.

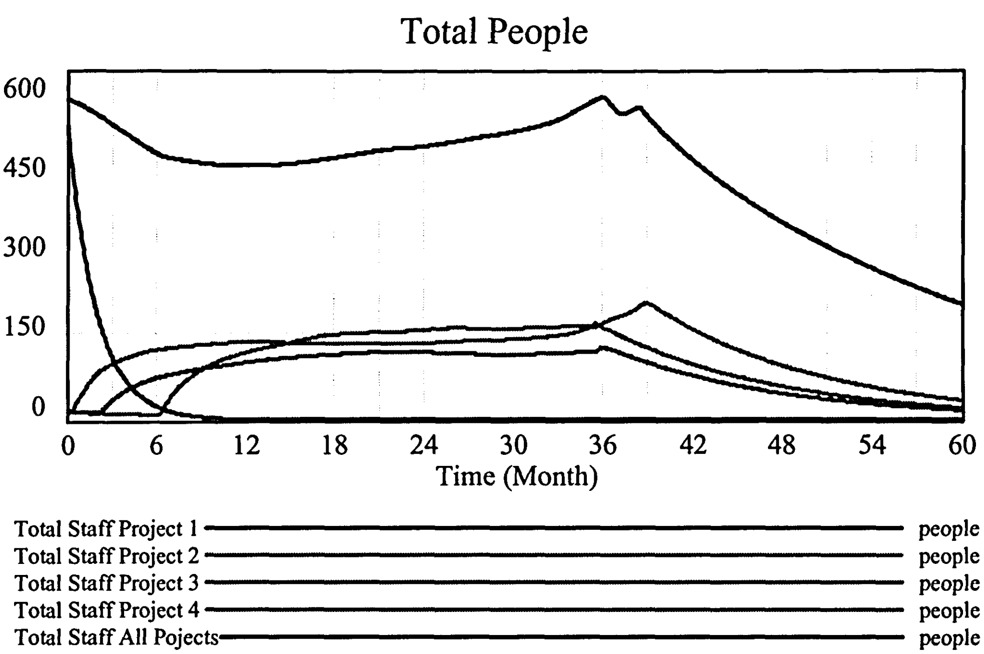

Figure 6-1 : Scenario A Staffing Profile

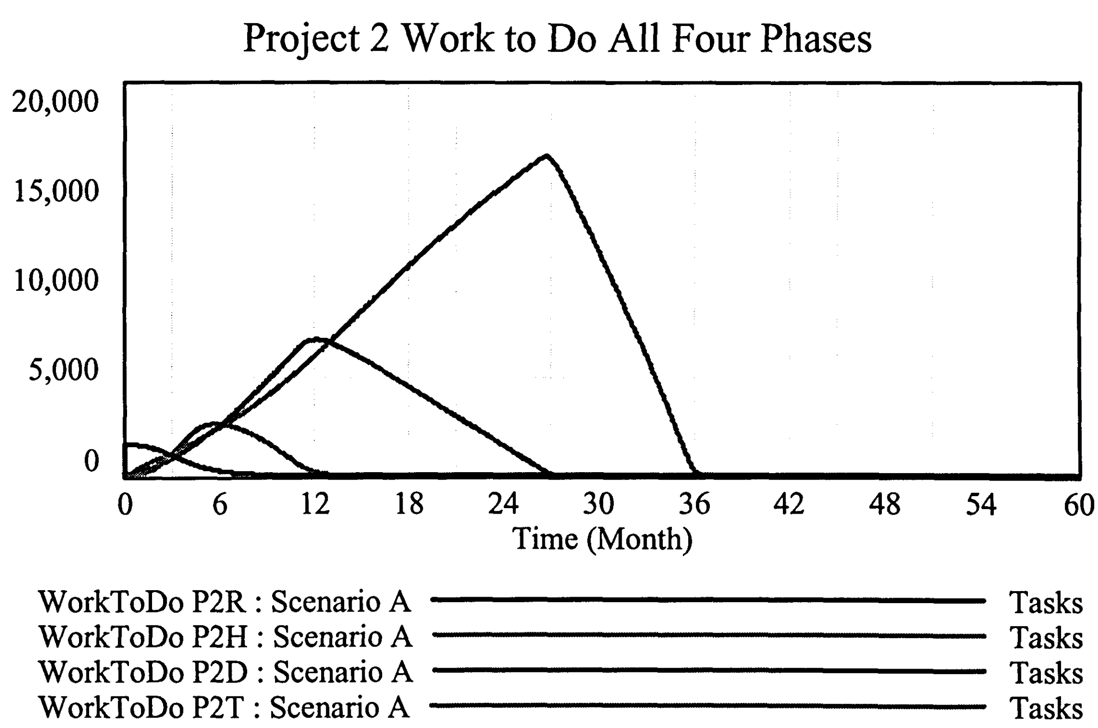

Figure 6-2 : Scenario A Project 2 Work To Do Profile

To recalibrate the model several changes were made to the base functions that are intended to capture real world experience. The Herweg-Pilon model implemented formulations that were divided into two categories: formulations related to work and formulations related to workforce. The formulations that were changed to calibrate the model are shown in Table 6-6. Discussions on how the baseline model was calibrated in relation to these formulations will follow.

Work	Workforce
Productivity • Effect of Fatigue - Not changed • **Complexity Effect - Changed**	Effectiveness • Novice to Intermediate Ratio Effect -Not Changed • Intermediate to Expert Ratio Effect - Not Changed
Quality • Effect of Fatigue - Not Changed • **Complexity Effect - Changed**	Learning • Staffing Gap Effect - Not Changed • Complexity Effect - Not Changed
Overtime • Overtime Effect - Not Changed	Hiring • Gap Effect - Not Changed
Attractiveness • Bug Ratio Effect - Not Changed • Complexity Effect - Not Changed • Priority Effect - Not Changed • Staffing Gap Effect - Not Changed	Attrition • **Fatigue Effect - Changed** • **Complexity Effect - Changed**

Table 6-6 : Model Formulations for Work and Workforce

6.1.2 Definition of Model Dynamics that Represent Current Company Practices

The key effects that are discussed in this thesis are effects on work due to productivity and quality. The key initiatives implemented by the company are intended to reduce cost and reduce schedule uncertainty are centered on initiatives that enhance quality and productivity. Overtime effects on work are assumed to be similar to that modeled by the original developers and there was no effort to change the parameters relating to overtime. All projects will demand overtime once the schedule is in jeopardy. Overtime is the first line of defense for reducing schedule risk and is used extensively during the product development process. The effects of fatigue and low morale due to overtime are evident at the company. The assumptions made in the Herweg-Pilon model were adequate to drive the model dynamics.

The attractiveness functions of the projects were also not changed. Project priorities were set equal for all of the programs. Management believes that all projects must be executed in order to keep all customers happy. The dynamic in evidence in the model is that the earlier starting projects obey the possession rule. Once a project is staffed it is difficult to poach people off the program. Major projects can poach staff from other projects if they are beset with a major problem, but the normal course of the product development process for all programs is to have at least one major problem. Therefore it is difficult to justify that one project is in more need than another project. Since all of these projects have a similar timeframe the attractiveness will be the same during the different phases.

All projects are required to fully develop a newly designed engine and the complexity for each project is set equal. The model will not elevate one program over another due to complexity. Initially the model is set up so that the projects are of low complexity. This is representative of the way the programs are run. The amount of work defines the staffing requirements; the complexity of the programs will not drive moving people from one project to another.

The model increases the attractiveness of a project as staffing levels fall below the required threshold to complete the project on time. This dynamic is a realistic representation of company dynamics since staffing will be managed more aggressively if there is a large shortage of manpower required to make schedule. At the company this is probably the biggest driver or moving staff from one program to another. The time to move people is fairly long since the company is large and highly bureaucratic. Finally, programs are penalized financially by the customer for poor performance. Due to the penalties the program will run over budget. The poor financials will cause the program to receive less support since there will be no budget to cover the added workforce.

6.1.3 Changes to Calibrate the model

The following sections will describe how these functions are applicable to calibrating the model to the current system performance

The key workforce effects are effectiveness of the staff, learning, and hiring. Attrition was modified for this model since the attrition levels were set extremely high in the original version. The high levels were not representative of attrition at Pratt & Whitney. While attrition has been high recently much of the attrition is due to moving a large amount of the workforce from Florida to Connecticut. The effect of fatigue on attrition function originally ramped up to 10 times the normal level as fatigue was accelerated to levels that were twice that of normal fatigue. The curve was changed to 2 times the normal level as fatigue achieved twice the normal level. Overtime is currently high and attrition at the company is more represented by the modifications to he curve.

6.1.3.1 Productivity

No changes were made to the functions that effect productivity. In the original model version productivity was effected by fatigue and project complexity. The fatigue effect on productivity is shown in Figure 6-3. As fatigue ramps to twice the normal levels experienced during the product development process productivity is reduced to 60% of normal productivity. In the case of this model fatigue is driven by the amount of overtime required. At the company, significant overtime is normally required during projects to keep the process on schedule. Overtime typically is required fairly constantly once the detail design phase begins.

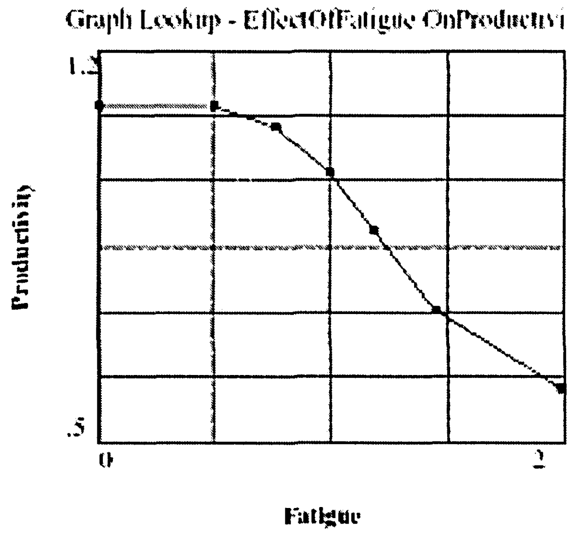

Figure 6-3 : Effect of Fatigue on Productivity

For the baseline case, there was no complexity effect on productivity. The model function assumed no degradation in productivity if the project was of low complexity. The baseline case assumes low complexity. The low complexity is assumed because of management belief at the company that there is no degradation in performance based on complexity. Therefore the plan is based on perfect execution. The staff can always handle all work at a maximum productivity level for the purposes of planning.

6.1.3.2 Quality

The baseline quality set in the model is assumed to be perfect. There is no initial degradation of the project since the projects are planned for perfect execution. Due to the short time schedules that have never been achieved in the history of the company and the low budgets allocated to keep product costs down the process is squeezed to the point that perfect execution has to be assumed in the beginning. The way the model drives the model quality is by using complexity. In the model, setting perfect quality requires the setting the complexity of the projects to one.

In the model, fatigue also effects the quality of work done by the employees. Employees make mistakes due to fatigue. In the model quality is greatly decreased as fatigue is increased from 1.5

times normal levels to 2 times normal levels. At twice normal levels quality will drop off to 65% of normal levels. At Pratt & Whitney programs do suffer from high levels of fatigue. The dynamic represented in the model is real.

6.1.3.3 Workforce

The systems dynamics model takes into account various effects on the workforce to account for training and mentoring of employees, skill level progression of employees, hiring and firing practices, and attrition. The effects of the workforce on the ability to complete the intentional iterations are represented in this section. The negative effects also cause more rework to be generated. The modifications related to attrition were addressed above. The processes and practices employed by the company are designed to help the workforce become more efficient so that less people may be employed to complete projects. The desire to keep the costs to a minimum by keeping the workforce to a minimum is the normal strategy employed by program management. The baseline case will help establish the dynamics of the workforce given the amount of work to do assuming perfect quality and productivity. Factors such as overtime and fatigue affect work and degrade performance in spite of productivity and quality gains made by the workforce and cause the requirement to increase the workforce. The sections below will discuss the effects on the project related to workforce.

6.1.3.4 Effectiveness

The model implements a productivity level associated with the experience level of each employee. The baseline case assumes that an experienced employee is three times more effective as a novice employee is and twice as effective as an intermediate skill level employee. Therefore the rate in which work is completed is dependent on the numbers of novice, intermediate, and expert level employees. Experience within the company over the years confirms that higher skill level employees are more effective and the assumption in the model is realistic. The baseline model was populated with slightly more intermediate skill level employees at the end of the initial programs than experts. The aerospace industry in general has suffered a decrease in the level of expert staff. Since the staff remaining in the company has come off the past programs they have progressed beyond novice skill levels, but have not moved to the expert levels. Employees who have worked multiple product development processes attain expert skill levels. There are no novices early reflecting the lack of hiring in the initial programs because of the anticipated decline in manpower requirements. The model also factors the effect of mentoring by expert and intermediate skill level employees. Normal levels of effectiveness are reduced if the ratio of intermediates to experts and novices to intermediates exceeds four-to-one. The effectiveness is greatly reduced as the ratio is increased beyond four-to-one. More employees that must be trained reduce the amount of work that a higher skilled employee can actually do. The trainer's time is spent training others and not accomplishing tasks. The non-linear effect was not modified in the model as experience training many new hire employees confirmed the response was real.

6.1.3.5 Learning

Skill advancement by training, either by class work or by on the job training, is highly important to creating an environment where productivity and quality of work are maintained at high levels during periods of hiring. The baseline learning time of a novice employee to progress to an intermediate employee and an intermediate employee to progress to an expert employee are set in the Microsoft Excel spreadsheet for model constants. For the baseline case this learning time is set to 24 months. The reason for the great amount of learning time is the high complexity of the product and the processes that must be learned. There are two factors included in the Herweg-Pilon model that can affect the baseline ability to learn of the employees. These factors are the staffing gap or shortage of people and the technical complexity of the projects. Learning is severely effected if there is a shortage of people to do the required work. The employees are so busy doing work that they can not learn new skills as rapidly as they would if they had time to study and learn about what they are doing. This effect was not modified in the model. As stated previously project complexity is set in the model constants and project complexity is taken into account for the programs. All programs are assumed to be of similar complexity. The Herweg-Pilon model was modified to remove the non-linear effect complexity had on learning since changes to learning time can be made by using the constants.

6.1.3.6 Hiring and Firing

The baseline hiring time is set within the model constant spreadsheet. The baseline hiring time is set to 8 months for the baseline case. Staffing shortages allow for the hiring time to be reduced by a non-linear function. In the case of overstaffing hiring is stopped. The maximum reduction in hiring time is set to half of the normal time due to system constraints. The baseline time to downsize is also set within the model constants. The model was changed in the baseline case to reflect a non-willingness to reduce workforce. The company is striving to keep the permanent workforce as constant as possible. Hiring is also affected by this policy. This is the reason hiring takes a relatively long time. The constants in the model were set up to damp out oscillations in the workforce. For these reasons understaffing of the projects is the normal procedure. Hiring of employees is a highly bureaucratic process in the company and typically takes over six months to obtain new hires once the need is expressed. The model time to hire was set to eight months. Current practice within the company is to avoid layoffs if possible. To dampen the system dynamic effects of the model and more represent the rate at which decisions are made to let employees go the model time to downsize constant was set to 24 months.

6.1.4 Scenario B - Baseline Project with Imperfect Quality, Productivity, and Complexity Effects

The difference between Scenario A and Scenario B illustrate the differences between planning for executing a project to perfection and executing a project with normal amounts of errors caused by human interactions in the process at lower than expected productivity levels. The following changes were made to the model and discussions of the reasons for the changes will follow.

Change	Description of Change
Complexity Project 2	1->10 (Productivity reduced by 15%, Quality Reduced by 20%)
Complexity Project 3	1->10 (Productivity reduced by 15%, Quality Reduced by 20%)
Complexity Project 4	1->10 (Productivity reduced by 15%, Quality Reduced by 20%)
Novice Skill Effect on Quality - Constant	1.0 -> 0.92
Intermediate Skill Effect on Quality - Constant	1.0 -> 0.95
Expert Skill Effect on Quality - Constant	1.0 -> 0.98
Maximum Quality in Concept Phase	1.0 -> 0.92
Maximum Quality in Preliminary Design Phase	1.0 -> 0.94
Maximum Quality in Detail Design Phase	1.0 -> 0.96
Maximum Quality in Validation Phase	1.0 -> 0.98

Table 6-7 : Scenario B - Changes Required to Reduce Productivity and Quality

Scenario B increases the complexity of the projects. Higher complexity has the effect of reducing baseline quality and productivity by 20% and 15% respectively. The productivity of the staff on the project will be degraded by problems introduced by the nature of the product. The design and production of a gas turbine engine is extremely complex as outlined in previous sections. Progress on completion of tasks rarely equals the expected performance at Pratt & Whitney. Productivity in the baseline case was reduced below perfect levels due to fatigue but without fatigue full productivity is possible in Scenario A. The difficulties in understanding all of the processes that are required develop the product drag down the productivity of the employees.

Baseline quality will not be as high as products that are easily designed and simple to assemble. The 80% quality level may still be high, but through the whole product development process it is close to the real number. The lower initial quality in the program is the reason that the quality initiatives are focusing on improving initial design quality. Poor quality affects the programs in two ways. More unintentional iterations are required due to the rework. To minimize the impact to rework risk mitigation plans are identified and put into place but most plans are developed after the initial program is completed. These iterations are unintentional in most programs.

Scenario B also introduces reductions in quality for novice, intermediate, and expert skill levels to demonstrate the effect of skill level on quality. A novice will be slower to complete a task than an expert employee will even though the expert's task may be more difficult. This change introduces differing levels of inefficiencies in the system. For each of the phases (Concept, Preliminary Design, Detail Design, and Validation), maximum quality was reduced to levels of 92%, 94%, 96%, and 98% for each of the phases respectively. The reason for reducing quality is the differing levels of abstraction and requirements definition at each phase. Early in the product development process the risk that requirements will change or be poorly defined are higher. The higher risks will reduce the quality of the work that is done in each phase. As the process further defines requirements the risk is reduced and maximum quality of the work can be raised. The levels that were chosen are the same as the original model constants set by Herweg and Pilon. The decision to use the values is intended to reduce quality slightly to see if the model dynamics are effected by poor requirement definition. The writer's knowledge of the product development process indicates that the assumption of lower quality work during each phase of the process is true. The assumptions will be evaluated to check for model sensitivity to these values. The data related to proving the validity of the value chosen could not be found. Changing the three projects to maximum complexity effects learning in the model. Learning will be slightly enhanced for the employees because complex projects are more interesting to the employee and introduce more issues that must be mastered. Faster learning will raise the quality and productivity of the workforce over time as the workers progress in skill levels.

6.1.5 Scenario C - Baseline Project with Imperfections and Project Instability

The additions in scenario C involve the Unknown, Unknowns. At Pratt & Whitney poor quality may generate rework and risk mitigation plans may have to be executed. These issues drag down the progress of a project by generating rework throughout the program. Some of the rework may even be delayed until after the program is finished the validation process. In addition to these issues there are always a few problems found after validation testing begins that generate a large amount of rework and were never anticipated. The rework involves re-testing and delays the program, which adds work to the process. Historical delays to the programs due to the lead-time to fix the problems uncovered late in the process are usually held to less than three months. These are the risks that first design quality initiatives attempt to mitigate. For the purpose of this model the assumption is made that there is a design problem found shortly after validation testing begins that drives significant rework

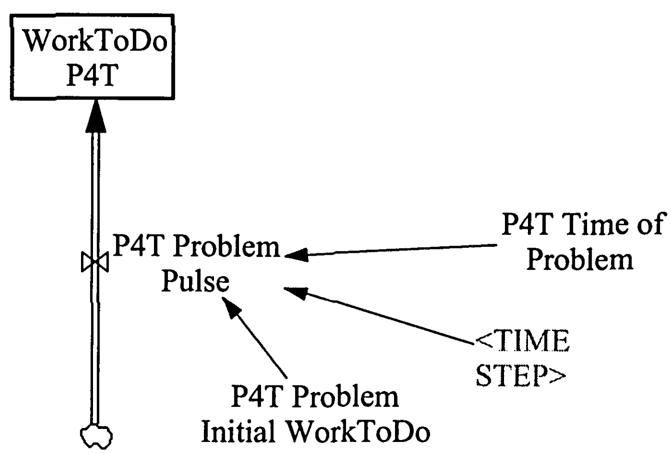

Figure 6-4 : System Dynamic Structure Added to Simulate Step Change due to Rework

The above structure was added to the Preliminary Design, Detail Design, and Validation phases of the model. The structure that can introduce a step change in the amount of work that must be accomplished in each of these phases. To add the amount of rework at the specified time the model was modified as shown in the table below. The rework added is intended to add approximately 3 months or less to the program end date.

Project	Validation Prob. Pulse Start	Validation Prob. Pulse Work
1	0	0
2	30	2400
3	25	2000
4	30	2000

Table 6-8 : Scenario C Constant Changes

The added work simulates the significant step change in work. These issues are implemented in the model to evaluate the effect of late instabilities on project cost and schedule risk.

6.1.6 Scenario D - Baseline Project (Imperfections and Instability included) with Productivity Improvements

The product development process is undergoing many enhancements outlined in this thesis which will enhance productivity. As outlined in the risk framework productivity is a key driver of risk for the product development processes to deliver a product within cost and schedule constraints. Typical program targets for improvements in productivity at Pratt & Whitney are approximately 15 %. For this reason the model will be studied with improvements set at 15% to study the ability of Pratt & Whitney to mitigate risk to cost and schedule with the planned productivity improvements. The complexity effect on the model can be changed to effect the overall productivity of the staffing. The first simulation changes the complexity effect so that productivity is raised by 15% to match the planned gains. The simulation is denoted by Scenario D Complexity 1.

Shortening the time for employees to advance their skills and produce work completion rates that equal the most highly skilled employees can also drive productivity enhancements. This idea was introduced during the description of the Herweg-Pilon system dynamic model's underlying assumptions related to intellectual capital. Better definition of the processes will aid in training employees and will improve the skill advancement of novice employees to intermediate employees as well as intermediate employees to expert employees. The background section described the process of standard work as being the method to define the processes at Pratt & Whitney. The second simulation tests the effect of improving the time to train employees. To more fully understand the effect of improving the time to advance the skill of employees the model was tested through an 80% range of improvement. Data from lean studies aimed at eliminating waste and use of enhanced technology tools indicate that 80% improvements in productivity are not outside the realm of possibility. Scenario D Learning Sensitivity outlines the model study that was performed relative to improving the learning time.

The last method to improve productivity in the model is to change the model constants for the normal productivity for each phase of the projects. Changing the normal productivity of the employees could improve the overall staff productivity beyond the 15% increase that was tested in Scenario D Complexity 1. A 15% improvement will not achieve full mitigation of cost and schedule, so the model will be tested with improvements to learning time and base staff productivity increased to levels that will bring cost and schedule metrics within the planned completion levels. The effect of decreasing the learning time to 20% of the original time is also included in an attempt to achieve cost and schedule goals. Scenario D Goal outlines the changes and results.

The following changes were made to the model to analyze the effect of productivity enhancements and will be discussed after the table.

Analysis Run	Change	Description of Change
Scenario D Complexity 1	Effect of Fatigue on Complexity function	Effect function changed to linear function from 1 to 10 so complexity does not reduce productivity by 15%. The number was picked to match the planned productivity improvement to the processes.
Scenario D Learning Sensitivity	Novice Advance to Intermediate and Intermediate Advance to Expert Time - Concept, Preliminary Design, Detail Design and Validation Phase	Sensitivity Analysis 4.8 months to 24 months (Scenario A). 80% Improvement in time to advance chosen to test the realm of outcomes based on possible improvement
Scenario D Normal Productivity Sensitivity	Normal Productivity Test Phase	Sensitivity Analysis 7 Tasks/(person*Month) (Scenario A) to 12.6 Months. 80% Improvement in productivity chosen to test the realm of outcomes based on possible improvement
Scenario D Goal	Advancement Time	24 Months to 4.8 Months
Scenario D Goal	Normal Productivity Test Phase	7 Tasks/(person*Month) -> 12.6 Tasks/(person*Month) (80% Improvement)

Table 6-9 : Scenario D Changes to Test Productivity Improvements

The productivity of the staff on the programs was multiplied by a factor of 1 due to the effect of complexity in Scenario A. In other words productivity was planned at completing a certain number of tasks per month per employee. Increasing complexity to 10 in Scenario B changed the effect of complexity on productivity by reducing productivity by 15%. The effect is simply an assessment of productivity levels not being as high as planned. Scenario D Complexity 1 assesses the effect of reversing the loss of productivity or that planned increases only improve productivity back to planned levels. The table for the complexity effect on productivity was changed to a straight line at 1 instead of the curve that reduced productivity as complexity was raised.

Scenario D Learning Sensitivity assesses the effect of reducing the time required for employees to become more skilled at their job. Increasing the skill level increases productivity of the employee as well as enhancing quality slightly. The model constants for the learning time were improved 80% using the sensitivity analysis function of Vensim. As described above the productivity enhancement is manifested in the effectiveness of the employee.

Scenario D Normal Productivity Sensitivity studies the effects of increasing the normal productivity of the employees at the different phases. Base productivity improvements in this model are demonstrated by improving the normal productivity of the employees. The model constants for normal productivity were improved by 80% using the sensitivity analysis function of Vensim.

Scenario D Goal implements the required changes so the model completes the programs to the level 1 schedule requirements. To achieve the schedule goals the full effect of all of the changes had to be instituted in the model. At the company it is hoped that standard work will define the processes and the processes can be streamlined through the use of the quality improvement tools and lean practices. The efforts in defining the processes by standard work and working to improve them by using quality tools are hoped to bring about these increases in productivity. Standard work will also aid in allowing lower skilled workers to learn job responsibilities faster by defining them rather than depending on tacit knowledge transfer only to communicate knowledge of job requirements. Classroom teaching in fundamental skills has also been instituted to help improve skill advancement at the company. The corporation as a whole pursues a strategy of promoting advanced degrees for all employees, which enhances the ability to advance in skill levels. Training time does cut back on productivity so there is a negative effect over the short term.

6.1.7 Scenario E - Baseline Project (Imperfections and Instability included) with Quality Improvements

As outlined in the background section of the thesis the quality improvement initiatives have the goal of improving the process and reducing defects. Productivity enhancements are a byproduct of using the quality tools to improve the process and use of new technologies to conduct complex experiments quickly and cheaply. The quality initiatives are also greatly instrumental in providing detail instructions on tasks that need to be accomplished, which helps to eliminate mistakes caused by poor definition. The Pratt & Whitney ACE process is similar to Six Sigma. The ultimate goal of these programs is to improve the process quality to 3.4 defects in 1 million opportunities. The scenario's tested reflect the intent of overall improvements in quality up to the six sigma level. The model simulations used to assess the effect of quality in the product development process are outlined below.

Analysis Run	Change	Description of Change
Scenario E 3 Sigma	Effect of Complexity on Quality function	Effect function changed so complexity of 10 reduces quality to 95.4% (3 Sigma) instead of 80% (Scenario B)
Scenario E Six Sigma	Effect of Complexity on Quality function	Effect function changed so complexity of 10 reduces quality to 99.9% (6 Sigma) instead of 80% (Scenario B)
Scenario E Six Sigma	Maximum Quality - Concept Phase	0.92 (Scenario B) -> 0.99
Scenario E Six Sigma	Maximum Quality - Preliminary Design Phase	0.94 (Scenario B) -> 0.99
Scenario E Six Sigma	Maximum Quality - Detail Design Phase	0.96 (Scenario B) -> 0.99
Scenario E Six Sigma	Maximum Quality - Validation Phase	0.98 (Scenario B) -> 0.99

Table 6-10 : Scenario E - Changes to Model to Improve Quality

The improvements in quality can be shown by improving the maximum quality in each phase of the process in the model constants table. As noted in the productivity section the model constant that drives baseline quality is complexity. In Scenario B quality was lowered by 20% when complexity was changed from 1 to 10. Since Six Sigma is currently the standard process improvement methodology the evaluations of the effects of quality improvement efforts will be in terms of improving quality to three and six sigma. Scenario E Three Sigma evaluates the product development processes if quality was improved from 80% to 95.4% or 66,807 defects for every 1 million opportunities. Scenario E Six Sigma changes the base productivity to 99% and also the maximum quality for each phase of the program to 99%. The difference in employees at each skill level degrades quality, and since the nature of the product development process is difficult to comprehend the levels of quality degradation was not changed.

6.1.8 Scenario F - Baseline Project (Imperfections and Instability included) utilizing Outsourcing

Outsourcing is being widely employed by the company to bridge the short term staffing requirements of the programs when the work to do is greater than the existing staff can handle. The company tries to avoid any layoffs and keeps a staff that will not have to be downsized during the programs. Outsource labor is planned to be of higher skill levels than new hires, but is also in limited quantity. Outsource labor can be brought in much faster than permanent hire labor, but there is some lag in the ability to hire and have full productivity unless the labor has already been trained to work at the company. The following structural changes were required to be made to the model.

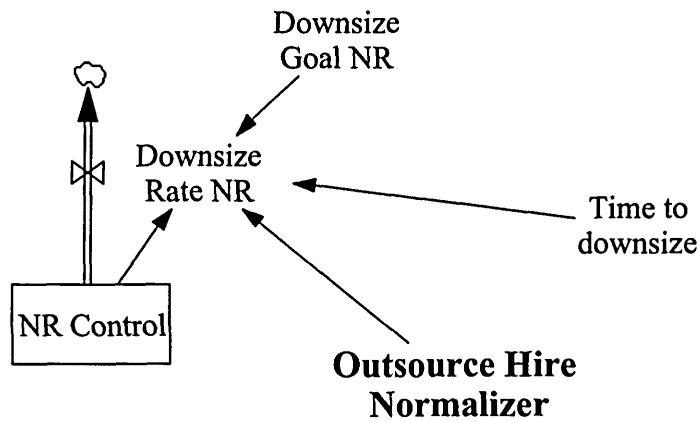

Figure 6-5 : Structural Changes to Implement Outsource Analysis

The changes in the model are outlined below and the discussion of the changes follows.

Analysis Run	Change	Description of Change
All Scenario F Runs	Novice Advance to Intermediate Time - All Phases	24 -> 1000 (Novice structure is used for outsource labor, no advancing)
All Scenario F Runs	Time To Downsize	24 (Scenario A) -> 1000, makes downsize rate low so manpower is kept constant. Outsource labor will be used to handle high workload
All Scenario F Runs	Outsource Hire Normalizer	1 -> 24, makes downsize rate equal to 1 so outsource labor can be shed immediately
All Scenario F Runs	Time to Hire Novices	8 (Scenario A) -> 2
Scenario F	Use of Overtime	Overtime use limited compared to all other scenario's. Only used in times of extreme need. Outsource labor planned to reduce overtime.
Scenario F	Novice Multiplier on Effectiveness	1 (Scenario A) -> 1.5 (Equivalent to Intermediate Skill)
Scenario F	Novice Skill Effect on Quality	0.92 (Scenario A) -> 0.95 (Equivalent to Intermediate Skill)
Scenario F Limit Hiring	Maximum Staff (Limit Outsource Hiring)	537 -> 400
Scenario F Limit Hiring	Novice Multiplier on Effectiveness	1 (Scenario A) -> 1.5 (Equivalent to Intermediate Skill)
Scenario F Limit Hiring	Novice Skill Effect on Quality	0.92 (Scenario A) -> 0.95 (Equivalent to Intermediate Skill)
Scenario F Best Case	Novice Multiplier on Effectiveness	1 (Scenario A) -> 3 (Equivalent to Expert Skill)
Scenario F Best Case	Novice Skill Effect on Quality	0.92 (Scenario A) -> 0.98 (Equivalent to Expert Skill)
Scenario F Best Case	Maximum Staff (Limit Outsource Hiring)	400 -> 537

Table 6-11 : Scenario F - Changes to Model to Enable Analysis of Outsource Labor Hiring

To evaluate the plans to use outsource labor the model was modified to remove the progression from novice to intermediate and all hiring was assumed to be done utilizing outsource labor. Productivity and quality of outsource labor is assumed to be at either expert or intermediate skill levels. For the purposes of this analysis one analysis was performed that changed the multiplier on effectiveness from a level of one for novice to three matching the expert multiplier. The multiplier affects the productivity of the employee. The effect of skill level on quality was changed to 0.99 to match the expert effect. Once this analysis was performed the numbers of outsource labor was found to be high. The availability of the outsource labor is an issue that would need to be addressed. The ability to hire skilled labor in the quantities that the model may try to drive to has to be limited. The model constant that controls maximum staff was changed from 537 to 400 to limit the hiring levels and check the sensitivity of the model to limitations in staffing. The last two analyses change the outsource labor skill levels from Intermediate to Expert with the maximum staff limited.

6.1.9 Scenario G - Baseline Project (Imperfections and Instability included) with lower initial utilization

The last policy test of the thesis is to look at utilization of the employees. Staffing at the programs is planned to be enough to cover the project as planned utilizing the people at 100%. Overtime is used to cover the problems introduced during the program.

Process management techniques use the concept that lead-time to complete work is the amount of time to actually complete the work and the time that the work must endure waiting for resources to become available. The waiting time can be a critical driver in the product development processes ability to complete work to do. Queuing theory shows that waiting time is gradually increased as more of a resource is used.[33] But as utilization passes 70%, delays can increase dramatically.

[33] Thomke, Stefan, "Enlightened Experimentation: The New Imperative for Innovation", Harvard Business Review, (February 2001), pp70

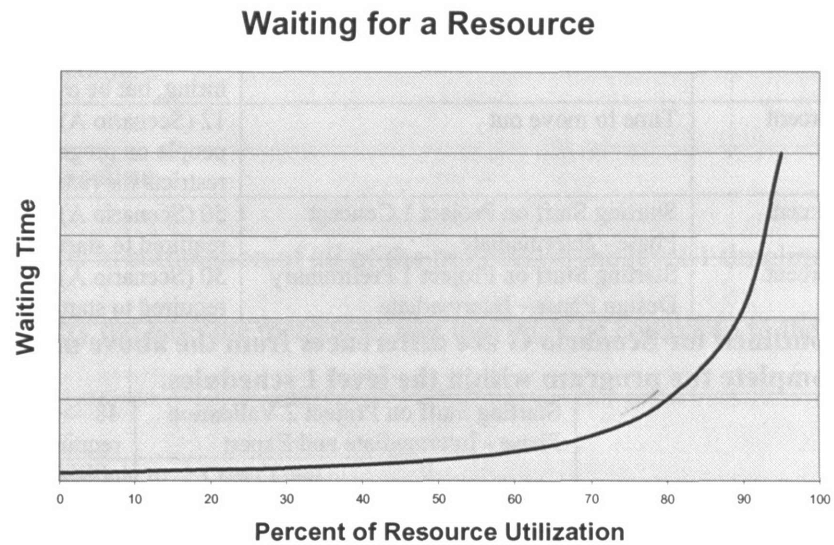

Figure 6-6 : Waiting Time Vs Percent of Resource Utilization

The last policy to be tested will start the program at 70% utilization rates and keep staffing changes to a minimum. The project will not undergo large amounts of hiring or firing. The workforce is intended to be stable. Changes outlined below will be made to achieve the utilization goals and constant staffing. If the project is not finished on time at the 70% level the staffing changes required for completing the project to the level 1 schedule requirements will be determined as the final Scenario G.

Analysis Run	**Change**	**Description of Change**
Scenario G 70 percent	Starting Staff on Project 1 Validation Phase - Intermediate	125 (Scenario A) -> 1 Transitioning staff from project 1 to other projects results in full staff not being transitioned to other projects as policy dictates
Scenario G 70 percent	Starting Staff on Project 1 Validation Phase - Intermediate	140 (Scenario A) -> 1 Transitioning staff from project 1 to other projects results in full staff not being transitioned to other projects as policy dictates
Scenario G 70 percent	Starting Staff on Project 2 Validation Phase - Intermediate and Expert	1 (Scenario A) -> 48 Amount of labor required to start project at 70% utilization
Scenario G 70 percent	Starting Staff on Project 3 Validation Phase - Intermediate and Expert	1 (Scenario A) -> 42 Amount of labor required to start project at 70% utilization
Scenario G 70 percent	Starting Staff on Project 4 Validation Phase - Intermediate and Expert	1 (Scenario A) -> 51 Amount of labor required to start project at 70% utilization
Scenario G 70 percent	Time for Attrition	12 (Scenario A) -> 100; Used to flatten curve of people leaving due to low utilization.

Scenario G 70 percent	Time to Downsize	24 (Scenario A) -> 1000; Policy is to not downsize, but to keep manpower
Scenario G 70 percent	Time to Hire	8 (Scenario A) -> 30; Policy is restrict hiring, but be overstaffed
Scenario G 70 percent	Time to move out	12 (Scenario A) -> 100; Policy is to keep people on program even if overstaffed, restricts movement.
Scenario G 70 percent	Starting Staff on Project 1 Concept Phase - Intermediate	50 (Scenario A) -> 20 Amount of labor required to start project at 70% utilization
Scenario G 70 percent	Starting Staff on Project 1 Preliminary Design Phase - Intermediate	50 (Scenario A) -> 60 Amount of labor required to start project at 70% utilization
The changes outlined for Scenario G are differences from the above model configuration required to complete the program within the level 1 schedules.		
Scenario G	Starting Staff on Project 2 Validation Phase - Intermediate and Expert	48 -> 85 Amount of labor required to start project at 70% utilization
Scenario G	Starting Staff on Project 3 Validation Phase - Intermediate and Expert	42 -> 65 Amount of labor required to start project at 70% utilization
Scenario G	Starting Staff on Project 4 Validation Phase - Intermediate and Expert	51 -> 75 Amount of labor required to start project at 70% utilization

Table 6-12 : Scenario G - Changes to Model to Analyze Lower Utilization

The key driver of risk that does not seem to have management plan is an adequate staffing plan based on the risk framework analysis. The literature review revealed that past research has indicated that low utilization of resources is key to remaining flexible and being able to handle instability in workload. To analyze this theory the model was exercised through initial staffing levels that are higher than the base levels. The utilization will be evaluated relative to the overtime variable. Staffing levels for Scenario G 70 percent were determined by finding the staffing level that produced an initial utilization of 70

6.2 Results

Implementation of the above modifications yielded the results presented in the following sections. The results will be presented relative to the ability of the policies to mitigate cost and schedule risk. Schedule is derived from the duration of the project and the risk is the relative slip in the program after problems are introduced and the risk mitigation strategy implemented. Cost is derived from the amount of labor that is required to complete the projects. Labor is measured in person*months and is typically a major driver in program cost. For the purposes of this thesis cost will be directly related to the cost of the programs. The following sections will also discuss the key drivers of the results relative to cost and schedule.

6.2.1 Scenario A - Baseline

The baseline case results in completion of all of the projects on the level 1 timeline. The following information will quantify the baseline metrics so that they may be compared to the policies that are being tested.

Project	Promised Level 1 Schedule	Schedule to Complete (Months)	Cost (Total Person*Months)
Project 2	35.5	35.6875	4527
Project 3	36	36	3454
Project 4	38	38.75	4635
Total			12616

Table 6-13 : Scenario A - Baseline Results

The duration metric is the measure of the ability of the projects to finish on time. In the baseline case all of the projects are assumed to finish within the promised level 1 schedule. The delta between the model results and the baseline case are caused by the model dynamics. The work to do tails off in completion rate at the end of the program due to people being removed from the program as the work to do drops off. A consistent point at the tail of the curve was picked to determine the completion time. When work to do fell below 700 tasks the project was deemed complete. The person*month metric is a measure of the baseline staff required to finish the project. The baseline case affirmed that much of the work to do is contained in the validation phase of the program for the group that is being examined. While testing the model it was found that all changes in the model relative to the concept phase, preliminary design phase, and detail design phase have a limited and negligible effect on the model dynamics. The reason for the negligible effect is the relative amount of work to do that is required in the first three phases. For this reason much of the model changes were applied only to the validation phase of the project. Data related to overtime, fatigue, productivity, and quality will be evaluated during the high work validation phase of the programs.

6.2.1.1 Overtime

During the planning of projects all programs expect to use overtime to handle the high workload periods that inevitably pop up during the program. Overtime is the most widely used method to keep projects on schedule. The plan is to only use overtime for short periods of time. The policy is easily implemented. The policy as practiced by the company does not provide compensation for the engineering staff and is also a cheap method to keep projects on schedule. The graph below shows that workload is high, overtime is maximized at 20 hours per week, as staff is brought onto the program to make up for staffing shortages early. The overtime plan shows how staffing is kept to a minimum and the employees are utilized at 100%, 40 hours per week, or more throughout the entire project. The significance of the graph is the level of overtime that is above 1. The rationale for maximum utilization is to reduce costs. Since the budgets are less than the amount project management deem necessary to complete all required tasks staffing levels are reduced to minimum levels to attempt to keep all tasks funded. Also, the baseline workload ramps up as the program aims to finish the program. A plot of staffing shows how the requirements for staff are greater towards the end of the program. The plots are shown below.

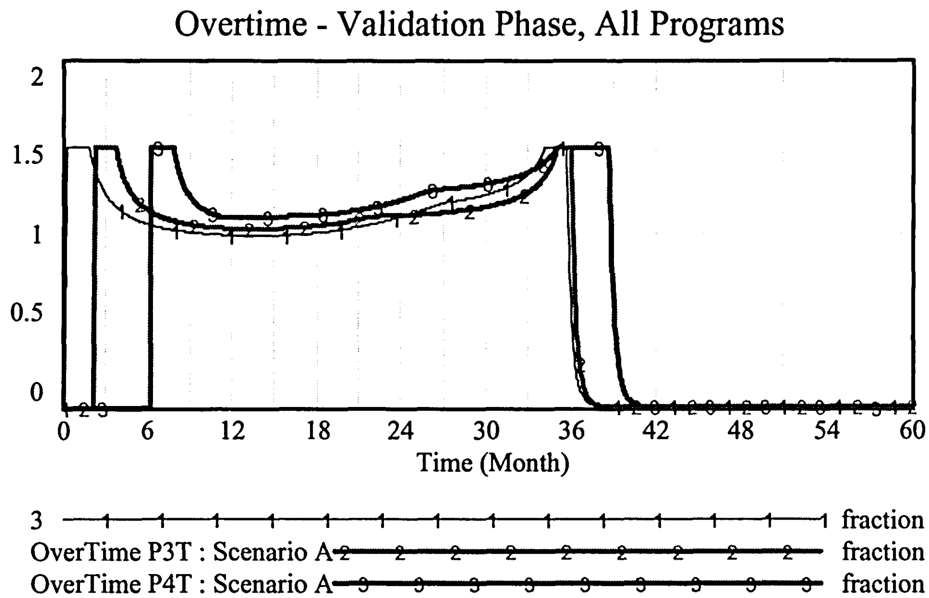

Figure 6-7 : Scenario A - Overtime Use on All Projects

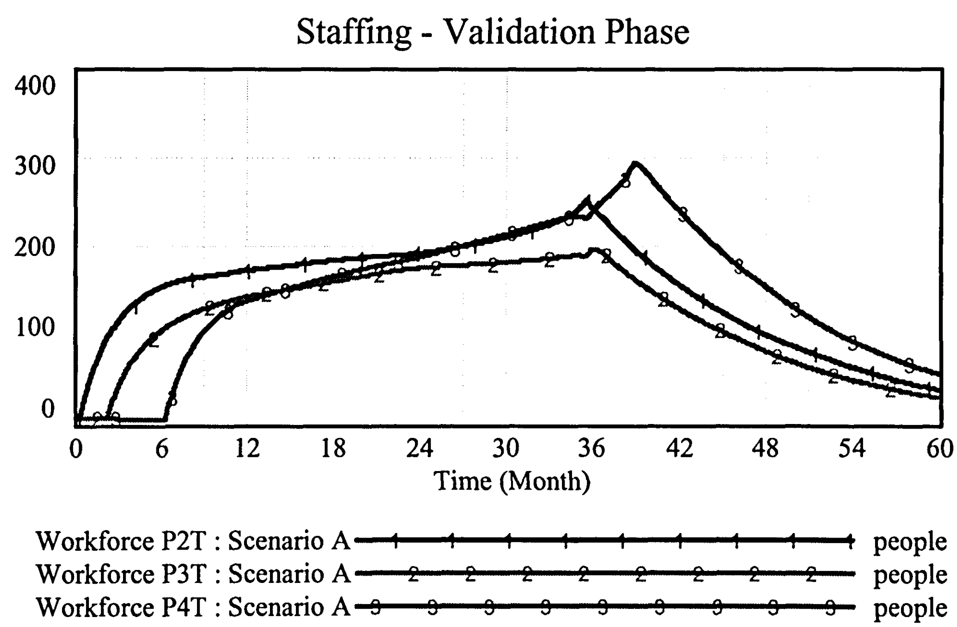

Figure 6-8 : Scenario A - Staffing Profile for the Projects

6.2.1.2 Productivity and Quality

Programs at Pratt & Whitney are planned for nearly flawless execution. For this reason Scenario A assumes high productivity and quality. The following graphs show the baseline quality and productivity.

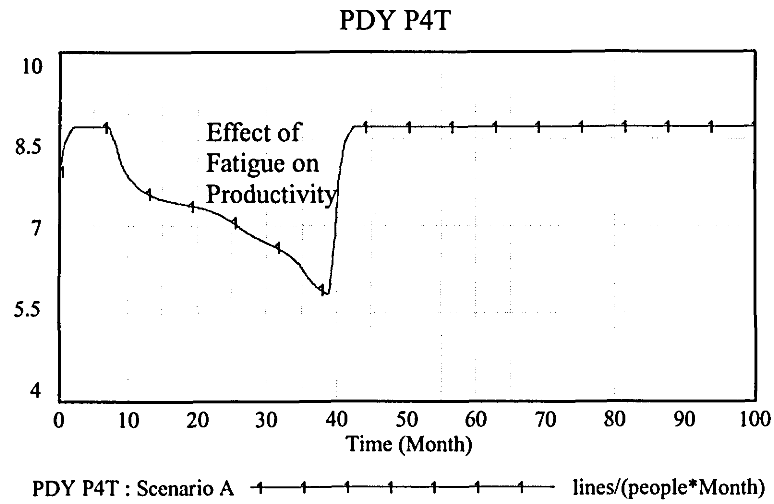

Figure 6-9 : Scenario A - Productivity on Program 4 - Validation Phase

Productivity is affected by fatigue in the baseline case. The project staffing is just enough to get the programs completed within schedule to reduce cost. Staff is required to work hard for the duration of the project and productivity suffers. Quality suffers from the same phenomena as shown in the following plot.

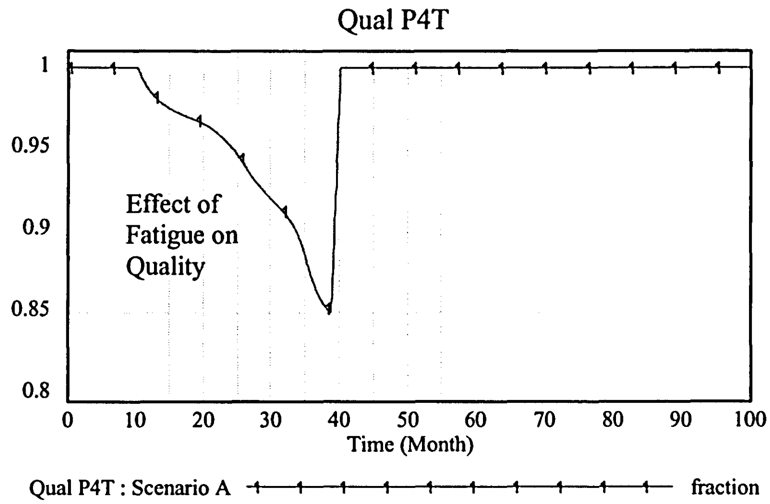

Figure 6-10 : Scenario A - Quality on Program 4 - Validation Phase

6.2.1.3 Summary

Scenario A represents the execution of the three programs according to plan. Staffing is planned to minimize cost and still complete the programs within the schedule constraints. The project completion times as promised to the customer and baseline costs as measured in people*months to achieve the promised schedule are summarized in Table 6-13. Staffing at these levels does result in some use of overtime, which degrades productivity and quality due to fatigue. The degradation in staff performance is not enough to compromise schedule. The utilization of the overtime policy is not detrimental to ability of the staffing to meet cost and schedule constraints. Overtime allows the existing staff to be able to complete the tasks required on time and within budget. The next section will explore the effects of the staff not performing at planned levels relative to productivity and quality and the resulting effects to cost and schedule.

6.2.2 Scenario B - Baseline Project with Imperfect Quality, Productivity, and Complexity Effects

As seen in Scenario A fatigue drives reductions in productivity and quality, but the programs are planned for near flawless execution. Scenario B introduces overall reductions of productivity and quality more representative of the reality of the execution of the projects. The overall effects of productivity and quality reductions are to greatly increase the amount of workers required to complete the projects. The table below shows the change from the baseline plan that programs suffer when quality and productivity are not as planned.

Project	Schedule (Months)	Schedule Slip (Months)	Delta % from Scenario A	Cost (Total Person*Months)	Cost Growth (Delta % from Scenario A)
Project 2	40.25	4.6	+12.8%	6969	+53.9%
Project 3	40.75	4.8	+13.2%	5334	+54.5%
Project 4	45.375	6.6	+17.1%	7493	+61.7%
Total/Avg		5.3		19796	+56.9%

Table 6-14 : Scenario B Cost and Schedule Differences from Baseline

Figure 6-11 shows the difference in staffing profiles when normal problems are introduced into the programs. The schedule slip has averaged 5.3 months relative to an average schedule length of 36.8 months or a 14.4% schedule growth over the course of the three programs. Schedule slips are the worst event that can happen in a program and all possible means to avoid schedule slip are attempted. Even though the number is relatively small the promises to the customer do not allow schedule slips to be tolerated. The cost of the program has increased 57% and is much higher relative to the schedule slip. The programs add staff in the drive to complete the program on time. The added staffing causes more new hires to be introduced into the system. As outlined in the description of the system dynamic model, the effect of more new hires is to reduce productivity of the staff who must devote effort to training the new staff. The quality of work for the new hires is less than that of experienced workers so more rework will be introduced.

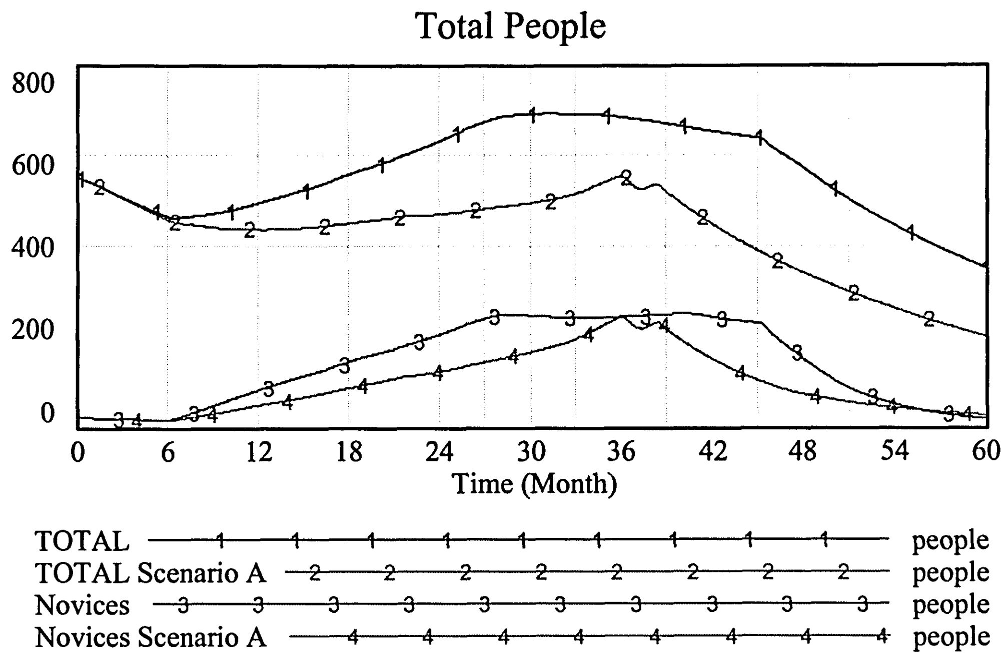

Figure 6-11 : Baseline Vs Baseline with Reduced Productivity and Quality

When comparing the model results to real programs at Pratt & Whitney the model shows staffing growth above the plan. Data from the project Z at Pratt & Whitney shown in Figure 6-12 demonstrates staffing growth above plan is real. The data on the left side of the graph shown in Figure 6-12 represents the real staffing requirements for project Z that has finished the validation phase. The right hand side of the plot shows the planned staffing for the project as a new development program enters the validation phase. The low levels of staffing in the middle of the plot are the real staffing requirements through the preliminary and early stages of the detail design phases. The plot demonstrates that the planned staffing does not equate to the real staffing requirements as projects progress through the product development process.

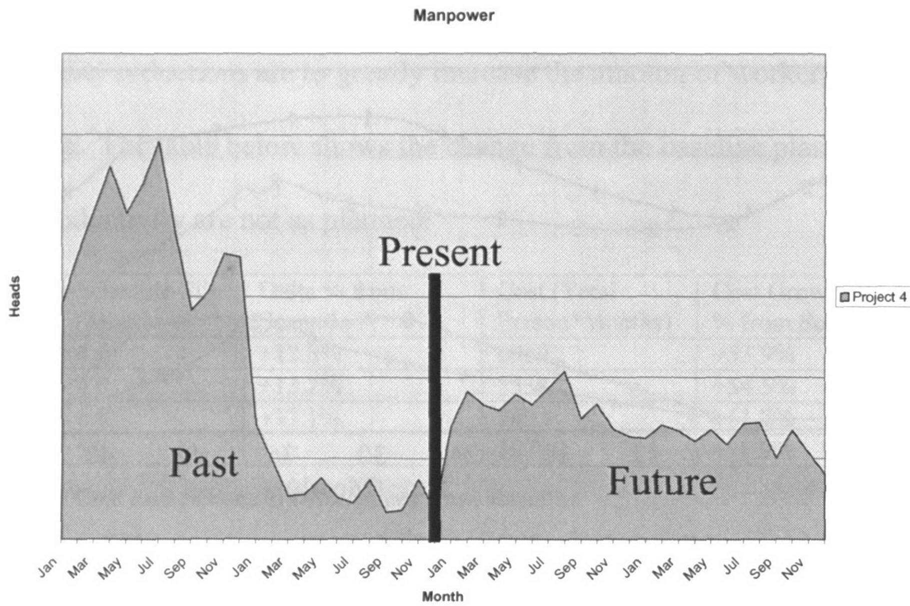

Figure 6-12 : Past, Present, and Future Staffing on Project 3

Further discussions of the project performance with problems introduced in regards to the overtime policy will be presented in the next section.

6.2.2.1 Overtime

Scenario B serves as a transition step to Scenario C. Scenario C will be the baseline realistic scenario that improvement policies will be measured against. The below analysis will serve to show how overtime that is required to keep the schedule constraint drives fatigue, which in turn drives further reduced quality and productivity. The graphs are intended to show trends of relative levels compared to the baseline project. As the project slips behind schedule the overtime is increased. Overtime drives fatigue up and quality and productivity are reduced. Figure 6-13 shows the how overtime is increased at the end of the program with the introduction of problems. The increased overtime is due to the system dynamics of the model. Staffing can not be brought up fast enough to handle the ramp in workload and the staffing that is brought on creates more rework since their quality levels are low due to inexperience.

Figure 6-14 gives the corresponding increase in fatigue, will drive the decreased productivity and quality demonstrated in scenario A. Decreased productivity slows the amount of work that can be done while decreased quality introduces more rework.

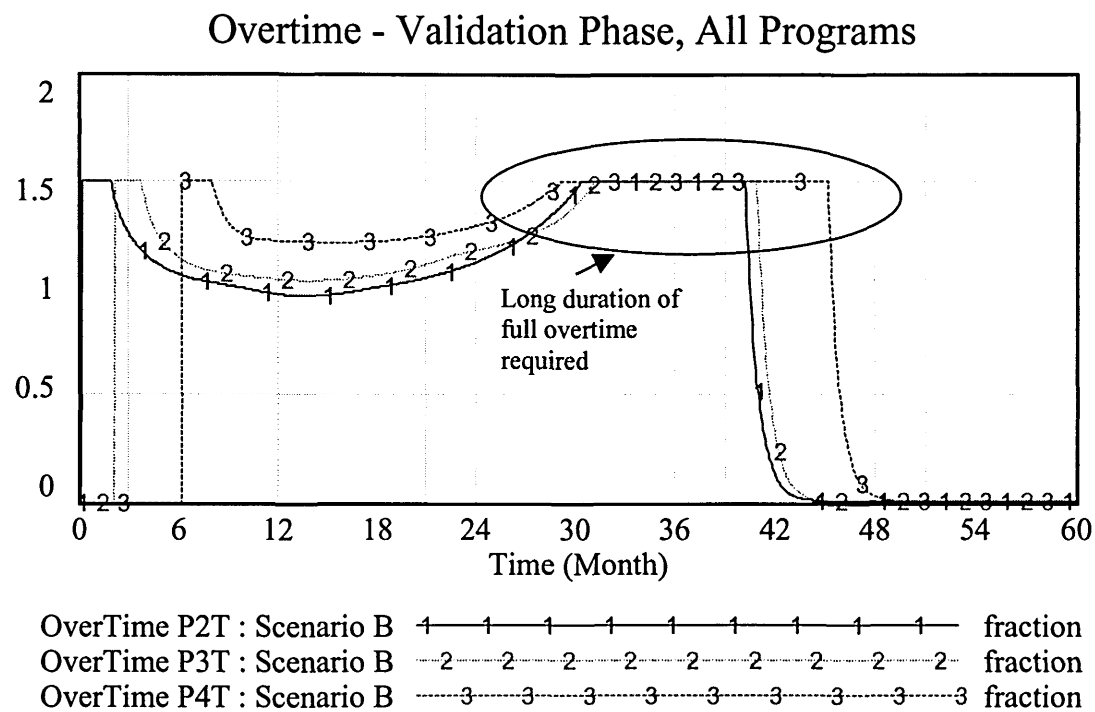

Figure 6-13 : Scenario B - Overtime increased in Level and Duration

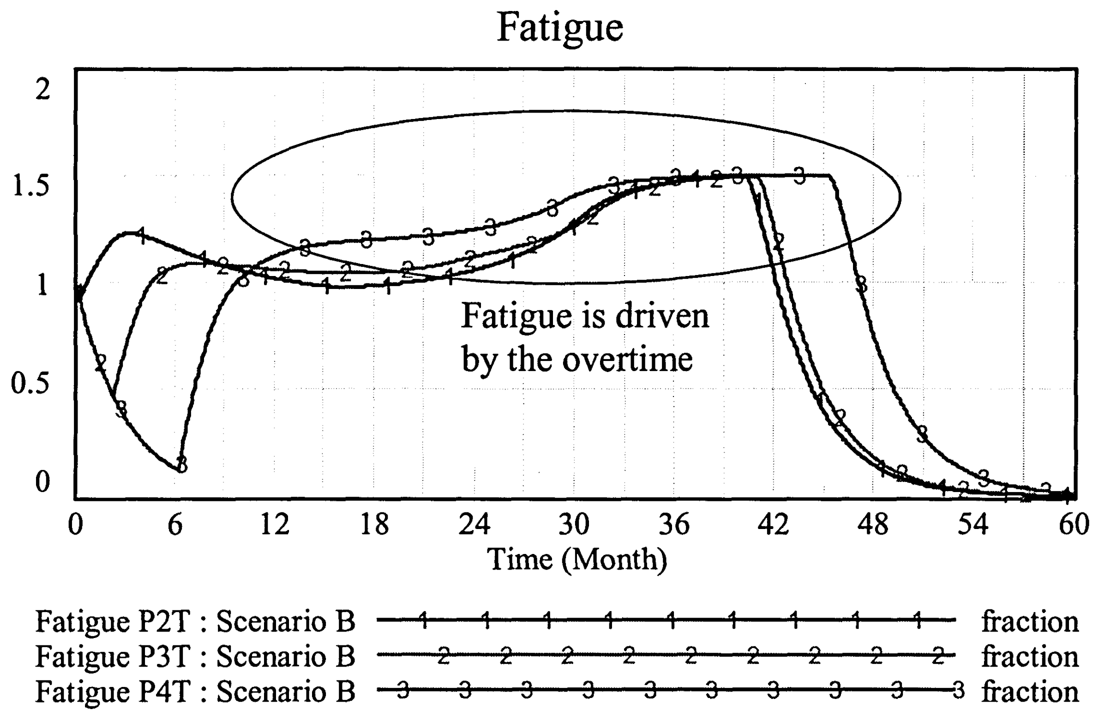

Figure 6-14 : Scenario B - Fatigue in Projects

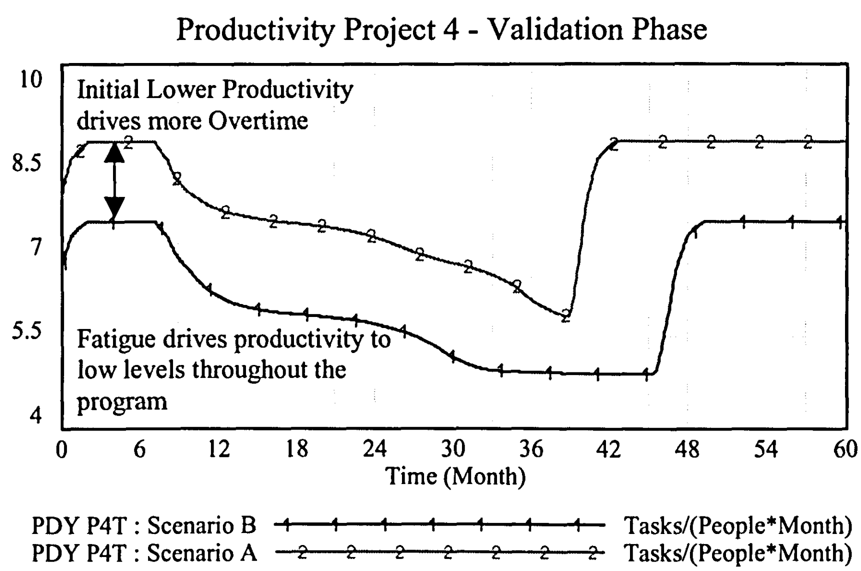

Figure 6-15 : Scenario B - Overall Productivity Significantly Lower than Planned

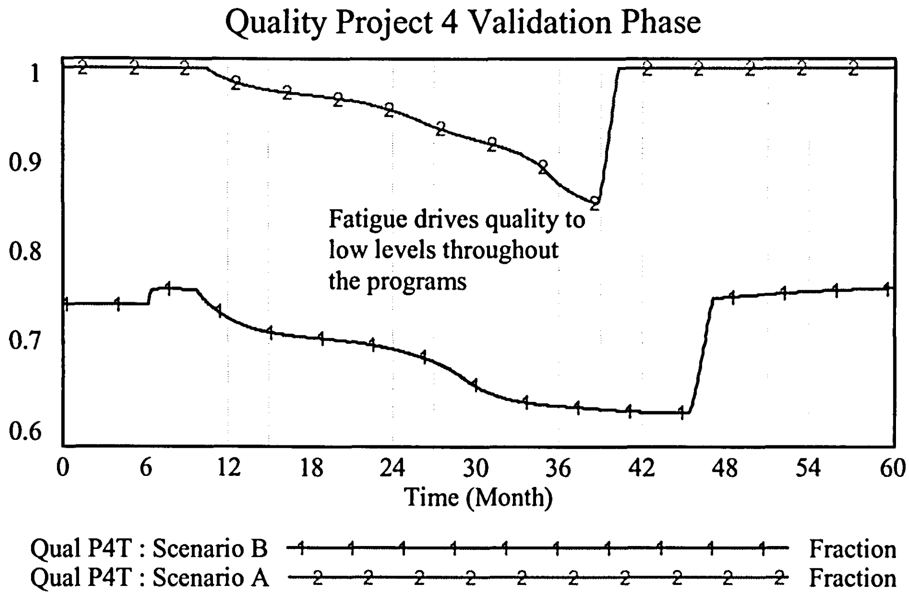

Figure 6-16 : Overall Quality Significantly Lower than Planned

Figure 6-17 shows the amount of additional rework or unintentional iterations that must be completed. The graph represents a 644% growth in rework on project 4 validation phase. The addition of significant amounts of rework is a large factor in the cost and schedule growth.

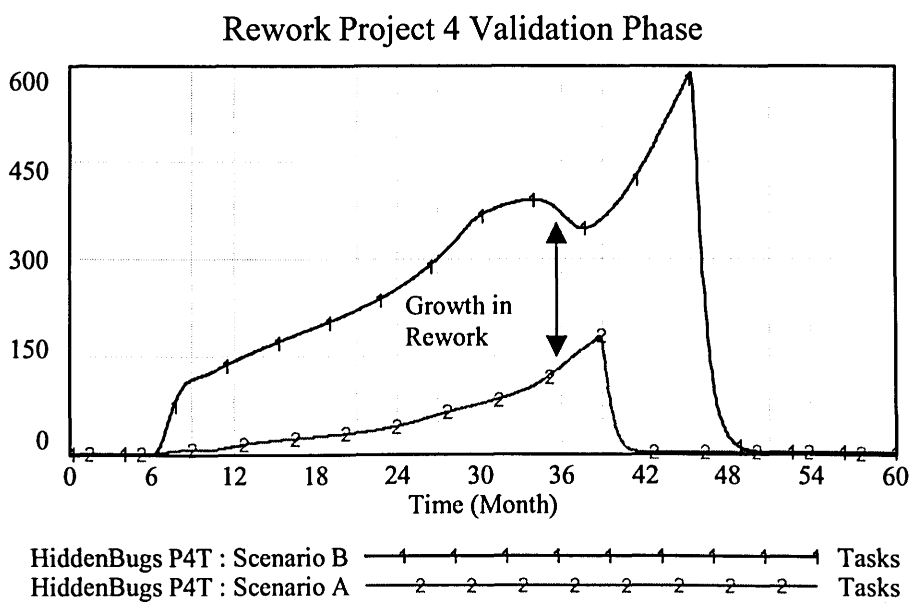

Figure 6-17 : Scenario B: Reduced Quality Increases the Amount of Rework

6.2.2.2 Other factors effecting performance

Due to the relatively large number of experts and intermediate staff there is no diminished effectiveness of the staff on the project because of mentoring and training of lower skilled workers. The resources are fairly well allocated across all programs due to the attractiveness being relatively equal. Some movement of manpower between projects exists due to staffing gap effects on attractiveness and completion dates as shown in the figure below.

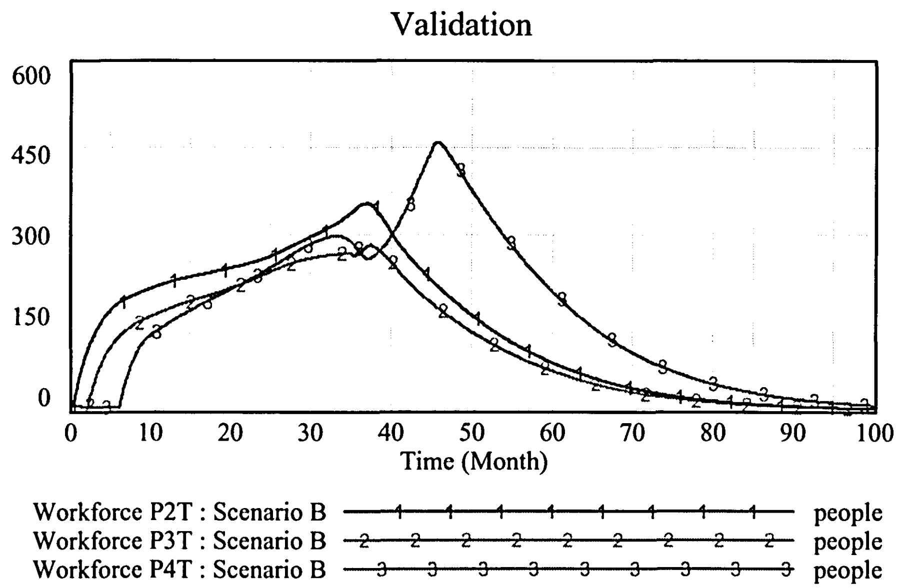

Figure 6-18 : Scenario B - Staffing Shifted from Project 4 in Months 32-37, Delaying Completion

Since the allocated resources are adequate to handle all three programs, no program will be completely ignored due to failures of another program. Project 4 is the last project to start. As projects 2 and 3 encounter higher workloads in month 33 they begin to poach resources from program 4. When projects 2 and 3 finish the workforce can be transferred back to project 4 to complete the project. The model dynamic is driven by the attractiveness of the program. Projects that start early get the required resources and also the more skilled resources. Later programs are understaffed, prone to poaching, and get lower skilled workers and will therefore finish late and over budget. The dynamics are representative of program dynamics at Pratt & Whitney. All programs must be executed to the level 1 schedule to satisfy the customer and avoid penalties. The program manager decides how much the program must try to get by with lower resources than required, but programs that are close to completion of the validation will steal resources from other programs as the pressure to flight test the product rises. The point is simply that the model is representative of what happens during multi-project execution if staffing levels are inadequate to complete all projects within schedule. The model dynamics correlate to real world situations.

6.2.2.3 Summary

The addition of 15% lower initial productivity levels and 20% lower initial quality than planned in the model results in a 5.3-month average schedule slip and a 57% average cost growth for the three projects. The use of overtime is the method that is used to manage risk in this scenario. The reduced levels of productivity and quality increases the use of full overtime to 9-15 months on the programs and because of the long duration of overtime fatigue diminishes productivity to half of Scenario A levels and quality to 64% of Scenario A levels. The low quality levels increase rework and low productivity levels slow progress getting the work complete. The use of overtime is ineffective in mitigating the cost and schedule risk because the length of use drives high levels of fatigue, which lowers quality and productivity. In this scenario overtime is not an adequate method to reduce risk to cost and schedule metrics. If productivity and quality do not meet the high expectations that the program plans are based upon the use of overtime is insufficient to complete projects on time and within budget. The flexibility that overtime can provide is fully utilized in the program as planned and can not mitigate the risks caused by rework and low productivity. The program assumption that overtime is an adequate policy to mitigate cost and schedule risks is proven to be incorrect. Scenario C will further evaluate the use of overtime to mitigate risk in the programs.

6.2.3 Scenario C - Baseline Project with Imperfections and Project Instability

A late program instability in workload is introduced into the validation phase of each of the three programs in Scenario C by adding work as outlined section 6.1.5 to the beginning of the validation phase. The instability that is introduced is within limits that most program manager's feel is within the scope of recovery within the given cost and schedule constraints. Higher levels of late problems will cause the program to replan the level 1 schedule since making up the schedule growth will be impossible within the constraints given. Figure 6-19 shows the added burden on the project cost and schedule with the late problems introduced. The policies will be evaluated using these model constraints. The table below shows the additional burden to the program.

Project	Schedule (Months)	Schedule Slip (Months)	Delta % from Scenario A	Cost (Total Person*Months)	Cost Growth (Delta % from Scenario A)
Project 2	42	6.3	+17.7%	7326	+61.8%
Project 3	42.3125	6.3	+17.5%	5624	+62.8%
Project 4	47	8.3	+21.3%	7817	+68.7%
Total		7.0		20767	+64.6%

Table 6-15 : Scenario C Cost and Schedule Differences from Baseline

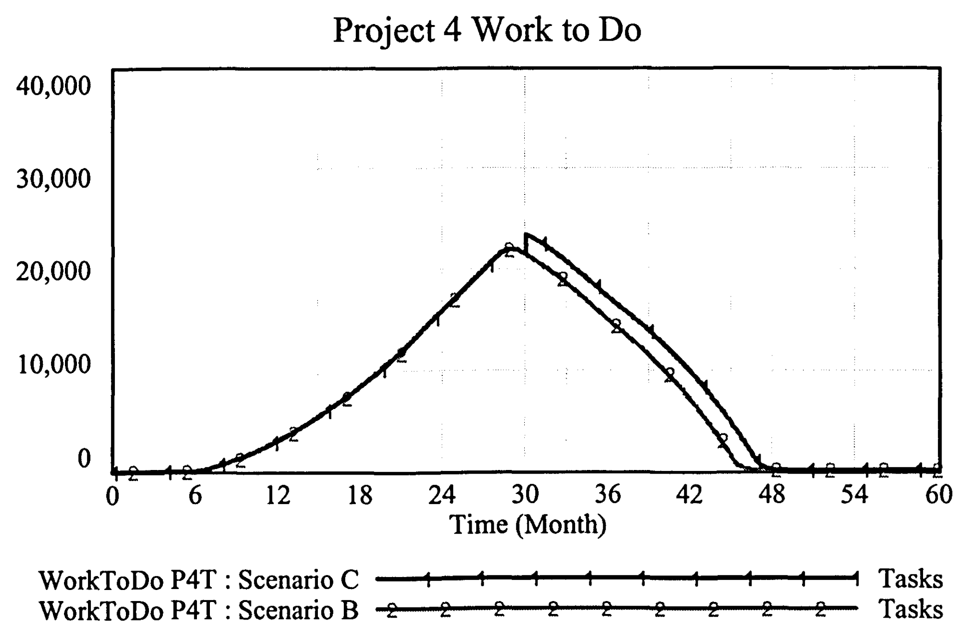

Figure 6-19 : Scenario C: Addition of Instability Adds 1.7 months to Schedule

Scenario C serves to introduce problems that are separate from the problems that are introduced by normal quality reductions. The lead-time associated with late problem causes a shift in the ability to conduct tasks as time for hardware redesign and procurement delays progress in the program. The final results indicating an average 7-month schedule slip and a 65% growth in program costs are very representative of the typical growth seen at the company. The model is consistent with reality at Pratt & Whitney. Policies to mitigate risk to cost and schedule must address the ability to handle these problems. The graph above shows overtime can not mitigate any risk to the program since there is no change to the curve of work done as the program finishes. Overtime can not be added since it is already at maximum levels. Since the plan has the program so far understaffed, there is no ability to improve the rate that work is completed. The program is at maximum output. At Pratt & Whitney by the time the program is in the latter stages of the program the work schedule has expanded to three shifts a day, seven days a week. Usually this schedule is implemented at the first occurrence of an unplanned iteration.

The metrics below show how overtime effects and problems that are normal to program execution degrade program performance. The rate at which work is done correctly is degraded and the growth in rework causes the schedule and staffing growth shown above. The effects of reduced quality, reduced productivity, and use of overtime drive the program performance level.

- Scenario A% Work Done Correctly at Peak Workload - 85.0%
- Scenario C% Work Done Correctly at Peak Workload - 63.2%

Project	Increase in Rework in Validation Phase from Scenario A
Project 2	1054.1%
Project 3	1104.7%
Project 4	662.2%

Table 6-16 : Scenario C Vs Scenario A Increase in Rework

The large percentage growth in rework is due to the lower quality that is a result of the combination of 20% lower than planned quality and the additional degradation of quality due to fatigue. This scenario is the reason that only 63% of the workload is done correctly. Figure 6-17 in Scenario B pictorially shows how the growth occurs.

6.2.3.1 Discussion of other factors effecting performance

There are other factors in the model that drive the dynamics. A discussion follows that evaluates these effects now that the model has been calibrated and the model dynamics relative to the policies ability to mitigate cost and schedule risk are required to be evaluated.

The effect of attractiveness of the earlier projects as described in Scenario B is seen in the usage of overtime in the different projects. Schedule and cost growth of the later program is the highest.

Project	Increase in OT Usage in Validation Phase
Project 2	322.5%
Project 3	512.5%
Project 4	715.7%

Table 6-17 : Scenario C - Overtime Increase in Validation Phase over Scenario A

Attrition in the model can also drive the dynamics of the system. The attrition rate differences between Scenario A and C were evaluated to determine if in the calibration of this model the dynamic had any effect on the outcome. The attrition rate in Scenario A was 10.5 % and in Scenario C was 10.8%. The attrition rate differences are negligible compared with the growth in rework.

Staffing gap has an effect on learning time in the model. The higher the staffing gap relative to the number of people required for the program the less learning that will occur as the existing staff is overworked. Learning time is degraded approximately 10% due to the staffing gap. The long time that is required to train the workforce negates this effect. The difference in rate of people progressing from one skill level to the next in any phase and in a given month during the projects is 1 versus 2 people/month. With 200 people in each phase of the program the number is very small.

The last effect that will be discussed is mentoring. The effect of mentoring new employees is still unaffected by the growth in new hires. The ratio of new hires to intermediate staff is not high enough to degrade performance.

These effects are realistic. In some areas of the company they may have a higher effect than in others, but overall the general trend is true to the operation. The following sections will not discuss model dynamics relative to these drivers.

6.2.3.2 Summary

The addition of a late problem drives an average additional 1.7 months of schedule slip into the program for a 7-month total slip. Cost grows an average of 65% for the three programs. The late problems occur because of the nature of the gas turbine engine validation process. The system effects can not be determined until the engines go to test and the long lead times keep engines from being built until relative late in the program. The drive toward achieving TRL 6 on the development programs is an attempt to alleviate the occurrence of late problems, which can have catastrophic effects on cost and schedule if they result in a long lead redesign and procurement of hardware. Scenario B already had a significant use of overtime and since fatigue is at maximum levels the additional work that is added can not be mitigated by more overtime. The schedule slips and costs grow in proportion to the added work. The staffing policy which plans for maximum use of personnel resources given a perfect plan is vulnerable to late changes in the program if productivity and quality through the length of the program is below planned levels. The reason is that overtime has already maximized the output of the staff and they can not work any faster. The results of this section are the baseline real world process performance metrics.

6.2.4 Scenario D - Baseline Project (Imperfections and Instability included) with Productivity Improvements

To analyze the effectiveness of productivity improvements the model was exercised through multiple runs that evaluated different types of productivity improvements. Improvements in direct productivity of the employees and average workforce productivity improvements due to faster progression of skill levels were the two methods employed to test improvements in productivity within the model structure.

6.2.4.1 Scenario D Complexity 1 - Complexity Function in Model Set Equal to One

The first Scenario increases the base productivity of the workers by 15%. As described above the 15% level was chosen due to the fact that Pratt & Whitney plans for productivity increases of 15% over the course of a project. The table below shows the ability of the increase in productivity to bring the projects back to the original cost and schedule requirements.

Project	Schedule (Months)	Schedule Slip (Months)	Delta % from Scenario A	Cost (Total Person*Months)	Cost Growth (Delta % from Scenario A)
Project 2	39.5625	3.9	10.9%	6081	34.3%
Project 3	39.6875	3.7	10.2%	4576	32.5%
Project 4	44	5.3	13.5%	6145	32.6%
Total		4.3		16802	33.2%

Table 6-18 : Scenario D Complexity 1 Cost and Schedule Differences from Baseline

The productivity improvement allows work to be done faster. The schedules are improved 2.5 to 3 months and the cost metric is improved by 50%. The increase in productivity results in a faster work rate that is the same as in Scenario A. Lower quality still exists in the system and rework is generated which adds to the work to do. Overtime is mandated as the schedule falls behind. The overtime requirement is not as high as in scenario C because the higher productivity levels of the staff allow the completion of more work faster. Therefore a slight reduction in rework from Scenario C is demonstrated.

	Scenario C	Scenario D Complexity 1	Improvement

Project	Increase in Rework in Validation Phase from Scenario A	Increase in Rework in Validation Phase from Scenario A	Improvement from Scenario C
Project 2	1054.1%	922%	12.5%
Project 3	1104.7%	1010%	8.5%
Project 4	662.2%	596%	10%

Table 6-19 : Rework is reduced from Scenario C

	Scenario C	Scenario D Complexity 1	Improvement
Project	Increase in OT Usage in Validation Phase from Scenario A	Increase in OT Usage in Validation Phase from Scenario A	Improvement from Scenario C
Project 2	322.5%	225%	30.1%
Project 3	512.5%	328%	36%
Project 4	715.7%	410%	43.1%

Table 6-20 : Less Overtime Required with Higher Productivity

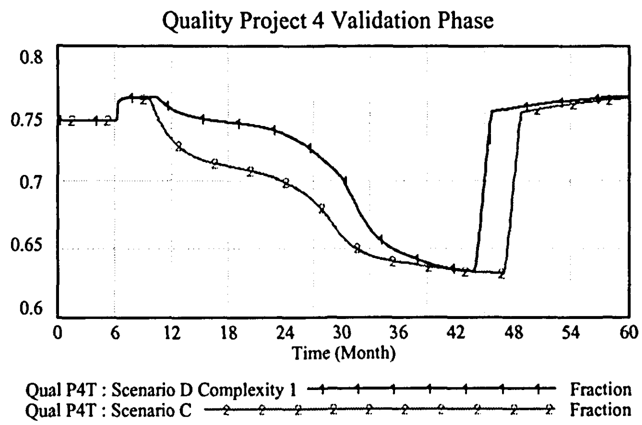

Figure 6-20 : Less Overtime Allows Higher Quality

Increasing productivity by 15% without increasing quality will not achieve cost and schedule goals. The amount of rework generated by lower than expected quality is as much as 9 times the expected amount. The improvement in rework is relatively insignificant (10%) compared with the Scenario C levels even though overtime use is reduced over a third. Improving productivity cuts down on the amount of rework but the lower than expected quality is more of a factor in driving rework. An overall 15% increase in employee productivity levels is not enough to overcome the 20% quality shortfall that is present in the model.

6.2.4.2 Scenario D Learning Sensitivity

The baseline case assumed the time to advance from novice to intermediate and intermediate to expert in all cases is 24 months. The definition of the advancement time is based on experience over the years training new hires in the company. To check the influence of reducing training time the model was exercised through a sensitivity analysis that reduced the advancement time in all cases to 4.8 months. As pointed out in the above section the amount of improvement was chosen to represent levels within the realm of possibility and to assess the ability of productivity improvements to mitigate the problems introduced into the product development process. The point of the exercise was to see if all skills at any level were matched to employees to provide the most efficient use of resources so that productivity could be improved to maximum levels. The improvement represents an 80% increase in productivity. According to a Lean Aerospace Report by Murman, Walton, and Rebentisch quantitative savings due to integrated product development processes were 80% less hours for design, 50% for NC Programming, 50% for inspection, and 67% for fabrication of flying testbed.[34] An 80% gain is not beyond possibility. Achieving these improvements in training time will require a concerted and structured effort to train above On-the-Job training. The corporation places a premium on education by spending $300M on education per year. A learning organization will in the end be a more skilled organization. The negative effect is that organized learning takes away from the productivity levels of the employees because the learning process distracts them. A full 80% improvement in reduction in learning is most likely a long-term proposition that will be unachievable in the life of the program. The point of the exercise is to check the sensitivity of the system to improvements in productivity.

Project	Schedule (Months)	Schedule Slip (Months)	Delta % from Scenario A	Cost (Total Person*Months)	Cost Growth (Delta % from Scenario A)
Project 2	37	1.3	3.7%	5772	27.5%
Project 3	36.875	0.9	2.4%	4269	23.6%

[34] Murman, Earll M., Walton, Myles, and Rebentisch, Eric, "Challenges in the Better, Faster, Cheaper Era of Aeronautical Design, Engineering, and Manufacturing", Lean Aerospace Intiative Report Series RP00-02, (September 2000)

Project 4	40.125	1.4	3.5%	5412	16.8%
Total		1.2		15453	22.5%

Table 6-21 : Scenario D Learning Sensitivity Cost and Schedule Differences from Baseline

With an 80% reduction in learning time for employees to progress from skill level to skill level through the use of extensive training the program schedule slip can be reduced from 7 months to 1.2 months. The cost risk to the programs is still 23% higher than the baseline. The reason for the improvement of project metrics is the faster progression of the workforce to higher skill levels. Higher skill levels result in:

- Higher effectiveness of the employee
  - Experts twice as effective as Intermediates
  - Intermediates twice as effective as Novices
- Less skill effect on quality resulting in slightly higher quality levels

The greater levels of higher skilled, more effective employee's result in significant reductions in overtime required in the program.

	Scenario C	**Scenario D Complexity 1**
Project	Increase in OT Usage in Validation Phase from Scenario A	Increase in OT Usage in Validation Phase from Scenario A
Project 2	322.5%	74%
Project 3	512.5%	93%
Project 4	715.7%	55%

Table 6-22 : Higher Average Skill Level of Employees Significantly Reduce Overtime

The significant reduction in overtime increases productivity and quality. Overtime that lasts longer than three months begins to degrade productivity and quality. Maximum use of overtime significantly degrades performance and can be seen at the bottom trough of the plots. The maximum level of overtime results in the lowest possible productivity and quality.

The time to get fatigued has been demonstrated at the company. Typically OT is used constantly, but not at maximum levels, i.e. work on Saturdays. Productivity and quality always suffer some during the use, but is much worse when it is used to operate 7 days per week, 3 shifts per day over periods that extend past a few months. Normally Monday can be spent fixing and redoing the work that was accomplished over a weekend for a variety of reasons.

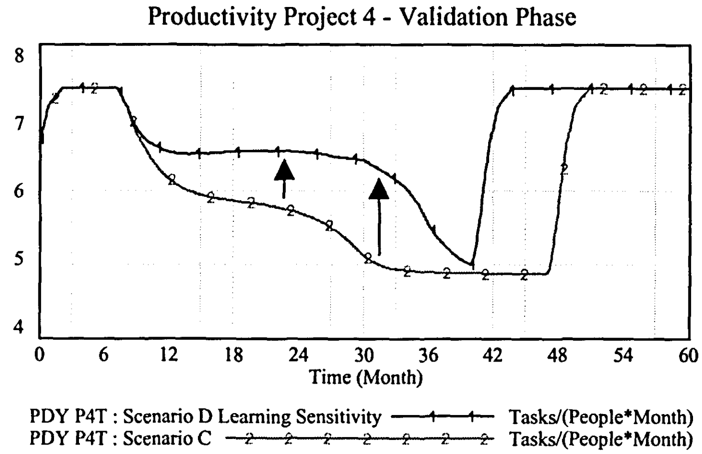

Figure 6-21 : Reduced Overtime with Faster Learning Increases Productivity

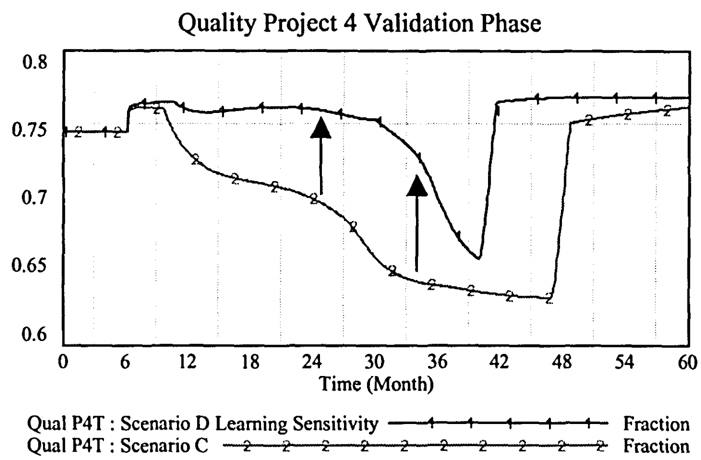

Figure 6-22 : Reduced Overtime with Higher Learning Results in Higher Quality

6.2.4.3 Scenario D Normal Productivity Sensitivity

Normal Productivity was varied between the base level of 7 and 12.6. The difference is an 80% gain in productivity. Processes would have to be significantly improved to result in gains this high. For example, typical development engine build times may take 3-4 months and the reduction would have to result in build times that were 1.5-2 months. Two to three week engine builds are stated goals for the validation group, but as yet have not been achieved. Increases in productivity that would result in higher gains than 80% are deemed possible, but to date very little productivity enhancement has been seen through the reorganization and process improvement effort. The table below shows that if productivity gains of this magnitude could be made the average schedule slip can be reduced from 7 months to 2.4 months and the cost growth is only 5.4% relative to 65% for the growth seen in the realistic Scenario C simulation.

Project	Schedule (Months)	Schedule Slip (Months)	Delta % from Scenario A	Cost (Total Person*Months)	Cost Growth (Delta % from Scenario A)
Project 2	38	2.3	6.5%	4808	6.2%
Project 3	38.0625	2.1	5.7%	3513	1.7%
Project 4	41.5	2.8	7.1%	4970	7.2%
Total		2.4		13291	5.4%

Table 6-23 : Scenario D Normal Productivity Cost and Schedule Differences from Baseline

Improving normal productivity shifts the overall productivity of the program. The table above shows that enhanced productivity has a significant effect on the cost of a program. More productive workers, even more productive than planned, can completely mitigate the cost risk due to lower than planned quality. The schedule risk has not been mitigated due to the high amount of rework that still must be accomplished. Productivity gains are useful in controlling program costs.

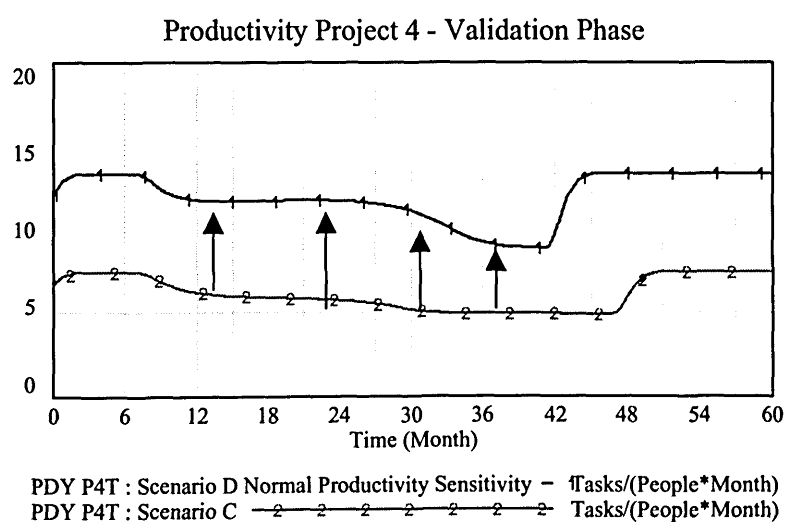

Figure 6-23 : Productivity Curve is shifted to the Level of the Improvement

The workforce is able to initially keep up with the tasks that are required to be completed. Approximately 60% less new hires are required to complete the tasks. The program does not need to bring on a high rate of new staff until late in the program

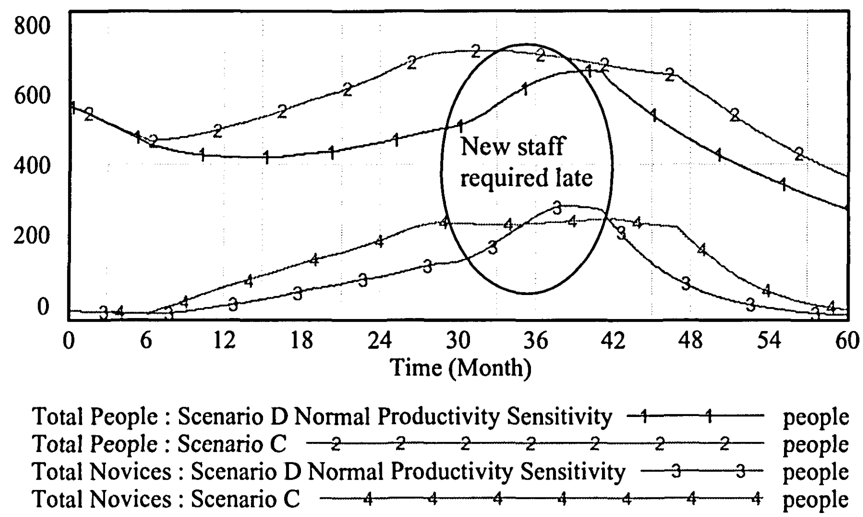

Figure 6-24 : Enhanced Productivity Causes hiring to be delayed to the End of the Program

The new hires are brought on at the end of the program. The increase in the amount of low skilled workers causes the generation of higher levels of rework at the end of the project. So, less rework is generated at the initial stages of the project, but more rework is generated later in the project.

	Scenario C	**Scenario D Normal Productivity Sensitivity**
Project	Increase in Rework in Validation Phase from Scenario A	Increase in Rework in Validation Phase from Scenario A
Project 2	1054.1	866%
Project 3	1104.7%	994%
Project 4	662.2%	544%

Table 6-24 : Scenario D Normal Productivity Vs Scenario C Increase in Rework

The result of the analysis is that productivity enhancements increase the rate at which work can be done. The increased rate means that less rework is generated relative to the amount of work that is being completed, but since a higher work rate is achieved similar amounts of rework are generated.

- Staffing equal between scenarios

- More rework is being generated at peak workload, but less is being generated throughout the bulk of the program.

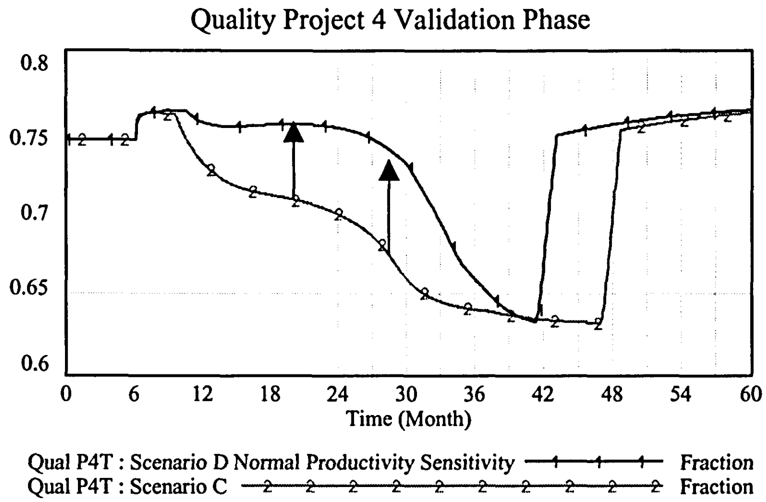

Figure 6-25 : Quality is the same at Peak Workload, but better Throughout Program

The graph above shows that quality is slightly higher in general throughout the program, but overtime still drives quality at the peak workload to the lowest levels seen in scenario C. The slightly higher quality drives the difference in rework seen in Figure 6-25. The rates at which work is completed for each of the Scenario's is shown below:

- Peak Rate at which work is accomplished in Scenario D Normal Productivity Sensitivity
  - 4560 Tasks/Month
- Peak Rate at which work is accomplished in Scenario C
  - 3375 Tasks/Month

These rates coupled with equal staffing between the programs results in the positive effects on cost and schedule.

Because of the higher productivity overtime is less significant than in Scenario C.

	Scenario C	**Scenario D Complexity 1**
Project	Increase in OT Usage in Validation Phase from Scenario A	Increase in OT Usage in Validation Phase from Scenario A
Project 2	222.5%	73%
Project 3	412.5%	123%
Project 4	615.7%	96%

Table 6-25 : Significant Overtime is still required on the Project

6.2.4.4 Scenario D Goal

The final scenario run to evaluate improved productivity is to change all of the productivity variables so that the programs can finish on schedule. The above analysis seems to indicate that a combination of improvements could drive the program to meet goals.

Project	Schedule (Months)	Schedule Slip (Months)	Delta % from Scenario A	Cost (Total Person*Months)	Cost Growth (Delta % from Scenario A)
Project 2	36.375	0.7	1.9%	3498	-22.7%
Project 3	36.3125	0.3	0.9%	2488	-28.0%
Project 4	38.6875	-0.1	-0.2%	3289	-29.0%
Total		0.3		9275	-26.5%

Table 6-26 : Scenario D Goal Cost and Schedule Differences from Baseline

The changes to the model to effect the result above increased the productivity of the individual employees 112%. In addition the learning time to go from novice to intermediate to expert skill workers was reduced from 24 to 4.8 months. In this scenario less overtime is required to complete the program than the baseline case presented in scenario A. The result is much less fatigue and hence higher quality. The percent of work done correctly at the peak workload is

- Scenario D Goal % Work Done Correctly at Peak Workload - 68.1%
- Scenario C % Work Done Correctly at Peak Workload - 63.2%

A highly efficient workforce that does not have to work much overtime can complete the project on time and under budget. Obviously 112% gains in productivity require quantum leaps in current productivity levels and are likely not achievable within the life of a program based on the progress thus far. The goal to achieve development engine builds that take less than one-month represent gains of 200% or more. To achieve the above goal requires a multitude of organizations to improve productivity level because they all feed the process of building an engine. The point of the company's reorganization and implementation of new processes and tools is to effect large gains in productivity.

6.2.4.5 Summary

Depending on the overall level of productivity improvements, schedule and cost goals of the program may be achieved. The typical program goals within the company are an improvement of 15%. Improving the base productivity levels of the employees by 15% only results in gaining back 2.7 months of the 7 month schedule slip due to lower than expected performance and late problems. Lower than anticipated quality in the programs causes 10 times the amount of rework and the added work can not be overcome with 15% gains in productivity.

But, lean manufacturing type gains of 80% or more in productivity levels could solve the program cost and schedule problems. If the productivity of the employees is raised 80%, the schedule slip can be reduced to 2.4 months and cost growth reduced to only 5.4% or less than typical management reserve levels. A smaller workforce will accomplish work much faster with high productivity. As the scheduled completion date approaches new workers need to be added to keep up with the higher than planned workload due to high quality, but the amount of rework caused by novice workers coupled with the increased fatigue due to overtime combine to slow the schedule progress.

Decreasing training time to 80% of the current 24 months, which enhances the average skill level and productivity of the employees reduced the schedule slip to 1.2 months but with a 22% cost growth over baseline. More highly skilled employees can complete the program with less overtime and fatigue levels. The cost and schedule growth is mostly due to initially low quality levels.

A combination of 112% productivity gains and 80% reduction in training time can completely mitigate lower than planned productivity and actually reduce costs by 26%. These goals are not likely to be seen within the course of the program, but they give the project manager an idea of the requirements for productivity improvements that will provide the ability to mitigate cost and schedule risk. The results indicate that successful management of the program can not rely solely on productivity gains.

6.2.5 Scenario E - Baseline Project (Imperfections and Instability included) with Quality Improvements

The above section demonstrated that productivity could overcome quality problems if substantial improvements can be made. In this section the thesis analyzes the ability of quality improvements to mitigate the risks to cost and schedule. Quality improvements are also a key strategy in the ability to manage the project to successful completion within cost and schedule constraints.

6.2.5.1 Improvement of Quality to Three Sigma and Six Sigma Levels

The following section will discuss the effect of improving quality in the programs. The results for cost and quality of the program relative to the goals are presented in the table below. The quality focus of the company is to improve quality to six sigma levels. The analysis looks at improving the quality of the programs to three sigma and six sigma levels. The results show that the project can get close to meeting the level 1 schedule with improvements in quality to six sigma levels, but the cost uncertainty is still high.

Project	Schedule (Months)	Schedule Slip (Months)	Delta % from Scenario A	Cost (Total Person*Months)	Cost Growth (Delta % from Scenario A)
Project 2	37.125	1.4	4.0%	5885	30.0%
Project 3	37.125	1.1	3.1%	4463	29.2%
Project 4	41.125	2.4	6.1%	6035	30.2%
Total		1.6		16383	29.9%

Table 6-27 : Cost and Schedule for Quality Improvement to Three Sigma Level

Project	Schedule (Months)	Schedule Slip (Months)	Delta % from Scenario A	Cost (Total Person*Months)	Cost Growth (Delta % from Scenario A)
Project 2	36.0	0.3	0.9%	5448	20.3%
Project 3	36.0625	0.1	0.2%	4150	20.2%
Project 4	39.6875	0.9	2.4%	5669	22.3%
Total		0.4		15267	21.0%

Table 6-28 : Cost and Schedule for Quality Improvement to Six Sigma Level

The quality runs are all very similar in effect. The slip in schedule is decreased from 7 months in Scenario C to 1.6 months for an improvement in quality from 80% to three sigma (95.4%) levels. If actual six sigma levels were possible to be achieved the schedule slip would be reduced to 0.4 months. Cost can not be mitigated because the worker productivity is still low and more resources are required the complete the project. For the above two cases costs overruns are reduced to 30% and 21% respectively. The analysis of the reasons for the added effort in the programs starts by evaluating the traces of quality.

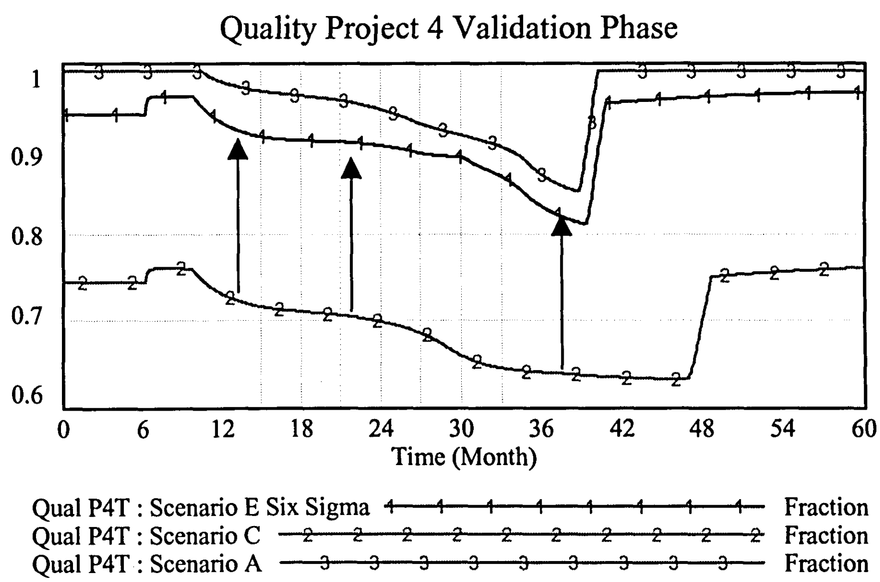

Figure 6-26 : Scenario E Quality Comparison with Scenario A and C

The chart shows that quality is significantly enhanced, but not quite to the levels of the baseline project.

- Scenario E Six Sigma % Work Done Correctly at Peak Workload - 81.2%
- Scenario A % Work Done Correctly at Peak Workload - 85.0%

The slight difference in the quality levels causes only 76% more rework than Scenario A. The amount of rework is significantly less than the rework levels seen in scenario C that were 10 times the amount of rework in Scenario A. The labor that is required to complete the project is still 21% higher than the amount required to complete the baseline program. The reason for this is the effect of productivity on the program. The amount of rework is significantly lower than Scenario C, but lower than expected productivity drives a need for more labor on the program than in Scenario A.

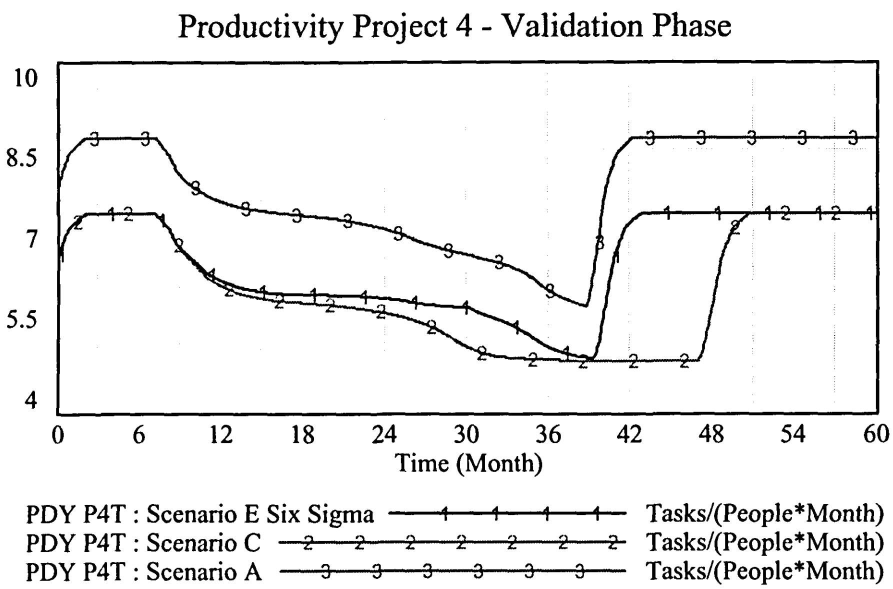

Figure 6-27 : Scenario E - Productivity Still Very Low Compared to Scenario A

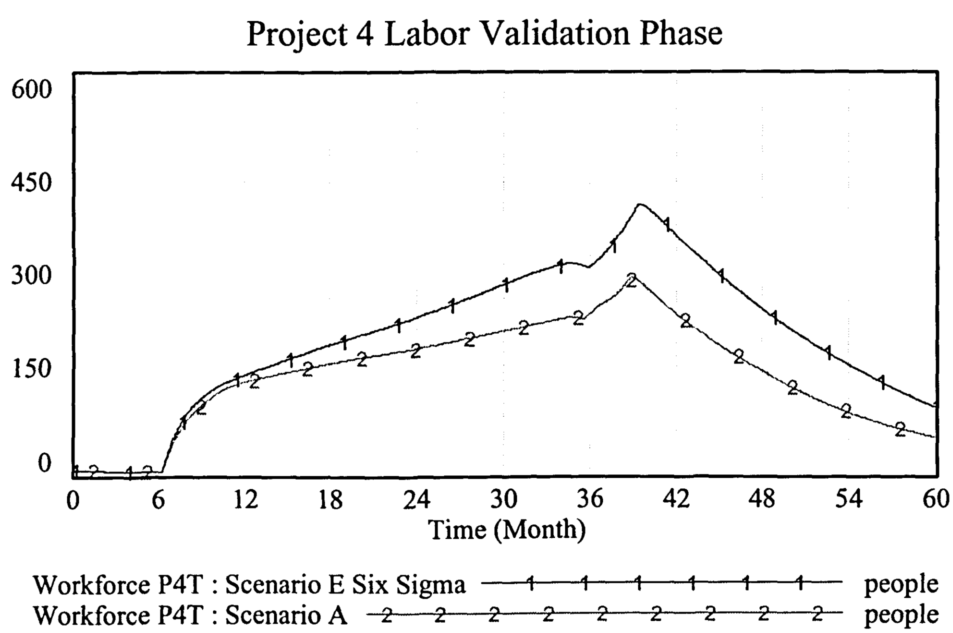

Figure 6-28 : Scenario E - Workforce Required Due to Lower Productivity is Higher and Drives Costs Up

Comparison of the rework at the baseline quality, three sigma, and six sigma levels shows that improvements in quality produce a linear improvement in rework levels. Therefore small improvements in quality will not yield a disproportionate gain in rework. Expectations would be that the curve would yield a lower ability to improve rework levels as quality approached perfection. The ability of quality gains to improve cost and schedule is not extremely powerful, but improvements can be made. The significance is the amount of cost that is required to achieve gains in quality grows exponentially as quality approaches perfection.

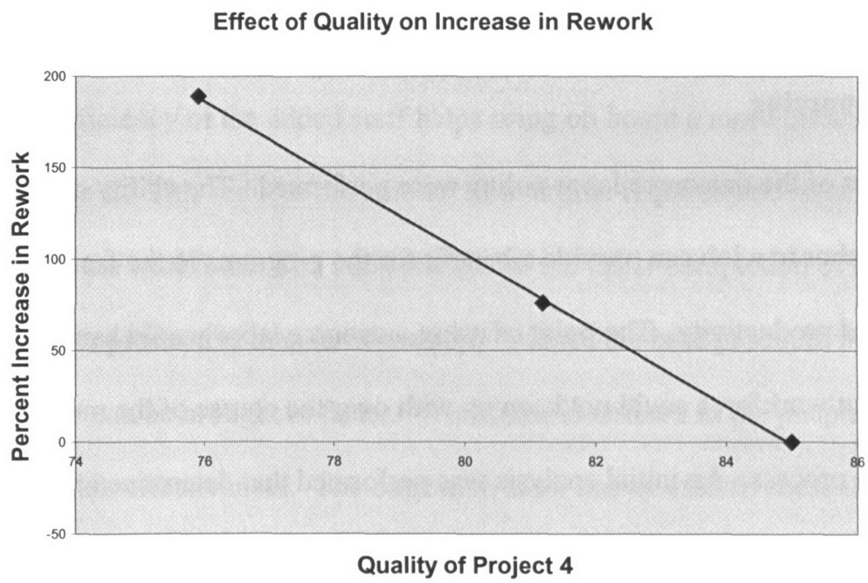

Figure 6-29 : Scenario E - Effect of Quality Improvements on Rework Linear

6.2.5.2 Summary

Quality improvement is central to the company philosophy. Quality initiatives attempt to drive quality levels toward six sigma or 3.4 defects in 1 million opportunities. Programs at Pratt & Whitney are planned such that all planned work will be done with little rework. Given this situation, if quality is initially at 80% and the program quality can be improved to three sigma levels or 95.4% schedule slip will be held to 1.6 months from the plan as opposed to 7 months. The cost growth of the program is almost 30%. Further improvement of the project quality to six sigma levels reduces the schedule slip to only 0.4 months with a program cost growth of 21%. The reason for the small slip in schedule is that the program adds people to cover the lower productivity, which increases the cost of the program. The schedule can be maintained because the rework levels are only 1 or 2 times higher for six sigma and three sigma quality levels as opposed to 10 times the level as seen in scenario C. High quality levels keep the work that is required to accomplish the program objectives closer to the plan. Lower than planned productivity drives the resource requirements higher and therefore costs are higher.

6.2.6 Scenario F - Baseline Project (Imperfections and Instability included) utilizing Outsourcing

Two analyses of the outsource labor policy were performed. The ability of outsource labor to quickly add skilled labor to a job can provide a benefit for the programs in the face of quality problems lower than expected productivity. The point of using outsource labor would handle a peak workload that the permanent workforce could not keep up with over the course of the multi-project product development process. An initial analysis was performed that determined that the resource requirements for outsource labor were too high and the model was modified to limit the labor. The first analysis following the initial analysis assumes the outsource labor comes in at Intermediate skill levels. The second analysis assumes the labor is of expert quality.

6.2.6.1 Initial Outsource Analysis

The initial analysis performed in Scenario F revealed that utilization of outsource labor could only marginally help growth in cost and schedule metrics. The schedule slip is only reduced in half from 7 months to 3.6 months. Cost growth has only been reduced from 64.5% to 46.6% from the realistic scenario.

Project	Schedule (Months)	Schedule Slip (Months)	Delta % from Scenario A	Cost (Total Person*Months)	Cost Growth (Delta % from Scenario A)

Project 2	39.0625	3.4	9.5%	6781	49.8%
Project 3	39.5	3.5	9.7%	5153	49.2%
Project 4	42.8125	4.1	10.5%	6564	41.6%
Total		3.6		18498	46.6%

Table 6-29 : Scenario F Outsource labor Cost and Schedule Differences from Baseline

If outsource labor is brought in at Intermediate levels the schedule can not be met and the amount of staff is still significantly higher than the baseline projects. The model does restrict the amount of people that can be brought into the programs. The amount of outsource labor that is brought in is similar to the amount of new hires that would be brought in to complete the projects in the above scenarios. Based on the above analyses relative to productivity and quality the results for outsource labor are similar because of the system dynamics related to productivity and quality.

The higher level of proficiency of the added staff helps bring on board a more effective workforce quicker than in Scenario C. This reduces the amount of overtime required and reduces the amount of fatigue experienced by the workforce. The main reason for the faster completion of the project is the larger workforce is more productive than in Scenario C because the base productivity of all workers hired is higher and they can be brought in faster. Workforce is defined as the people assigned to the project multiplied by their effectiveness. The outsource labor has an effectiveness that is assumed to be the same as intermediate skill labor, which is 1.5 times higher than novice labor.

The added staff produces work at only slightly higher quality levels than in Scenario C. The quality level is 0.95 as opposed to 0.92. The small enhancement in quality slightly reduces the amount rework that is required. This drives the amount of rework high and results in high level of overtime. Again, high levels of overtime increase fatigue and drive lower quality. The higher effectiveness and resulting higher productivity are the reason for the improvement over the scenario C results.

Schedule risk is not mitigated and cost risk is far from being mitigated because the workload is not significantly improved.

- Scenario F % Work Done Correctly at Peak Workload - 63.7%
- Minimum Productivity - 4.9 Tasks/Person*Month
- Amount of Outsource Labor Required - 348 people

- Total Labor Required on the Program - 705 people

6.2.6.2 Outsource Labor with hiring limits.

The fact that such a large number of outsource labor is required the model must be changed to cap the growth of labor so that the model is more reflective of the ability to hire outsource labor. The maximum people allowed in the Validation phase of the program was capped by reducing the amount of people allowed on the validation phase of the program. For all other scenarios 537 people were allowed to be on any one program during the validation phase. The maximum amount staff allowed was reduced to 400 so less outsource labor personnel could be hired. The results are shown below.

Project	Schedule (Months)	Schedule Slip (Months)	Delta % from Scenario A	Cost (Total Person*Months)	Cost Growth (Delta % from Scenario A)
Project 2	41.6875	6.0	16.8%	6916	52.8%
Project 3	41.9275	5.9	16.5%	5286	53.0%
Project 4	46.25	7.5	19.4%	6758	45.8%
Total		6.5		18960	50.3%

Table 6-30 : Scenario F - Utilize outsource labor, but limit hiring

- Amount of Outsource Labor Required - 260 people
- Total Labor Required on the Program - 660 people

The schedule slip relative to scenario C is now almost the same, 6.5 months relative to 7.0 months. The cost growth has been reduced from 64.5% to 50.3%. Utilizing a policy which attempts to handle peak workloads with outsource labor that are as effective as intermediate level staff is not an effective way to handle significant shortfalls in productivity and quality from planned level and added workload. Scenario F Best Case was run to see if finding expert skill level outsource labor in the same quantities as above would mitigate cost and schedule risk. The results of using outsource labor that has a higher skill level is shown below.

Project	Schedule (Months)	Schedule Slip (Months)	Delta % from Scenario A	Cost (Total Person*Months)	Cost Growth (Delta % from Scenario A)
Project 2	38	2.3	6.5%	5732	26.6%
Project 3	38.5	2.5	6.9%	4432	28.3%
Project 4	42.125	3.4	8.7%	5550	19.7%
Total		2.7		15714	24.6%

Table 6-31 : Scenario F - Utilize highly skilled outsource labor, but limit hiring

The reasons for the better ability to achieve cost and schedule goals are the same as adding staff at . intermediate skill levels. The productivity of the workforce is doubled over the previous scenario and quality of the added workforce is raised from 95% to 98% of baseline quality. The added productivity of the staff will reduce the amount of overtime required and further raise quality levels. Rework is reduced.

	Scenario C	**Scenario F**
Project	Increase in Rework in Validation Phase from Scenario A	Increase in Rework in Validation Phase from Scenario A
Project 2	1054.1	720%
Project 3	1104.7%	908%
Project 4	662.2%	531%

Table 6-32 : Scenario F Increase in Rework

The workforce commits less error, further improving the ability of the people to complete the programs faster. Outsource staffing level requirements are similar to the intermediate skill analysis.

- Scenario F Best Case % Work Done Correctly at Peak Workload - 66.7%
- Minimum Productivity - 5.1 Tasks/Person*Month
- Amount of Outsource Labor Required - 248 people
- Total Labor Required on the Program - 642 people

6.2.6.3 Summary

The analysis on the use of outsource labor shows that if hiring is limited to outsource personnel that is equivalent to intermediate skilled staff to handle peak workloads schedule slippage can be limited to 3.6 months or almost half of the original schedule. The amount of hiring in the model for the first part of the analysis was held to a similar amount as that in previous scenarios for comparison purposes. Cost growth is extremely high, 46.6%, and the reason is that program quality is still very low and base productivity has not been raised. In the case of this model the peak workload drives a high requirement for staffing. The requirement for staffing is as high as all the previous scenarios.

A problem that is brought up by this scenario is the company's ability to hire the quantity of personnel required is limited when restricted to outsource labor. The model was modified to assess the effect of limiting the ability to bring on high quality staff. Reducing the number of people hired by 88 people, or approximately a 25% reduction in the ability to hire, reduced the gain in schedule variance by only 0.5 months from scenario C to 6.5 months and reduced cost growth from 64.6% to 50.3%. The policy of hiring outsource labor to handle peaks in workload is limited to relatively low overruns in the planned work to be accomplished.

Further testing of the policy to use outsource labor policy by being able to hire the labor at expert skill levels, double that of the previous scenario, yielded better results. Cost and schedule growth over scenario C is reduced to 24.6% and 2.7 months respectively. The gain in cost and schedule is due to base quality levels still being at the 80% level. Rework levels are still high and the productivity gain by hiring more skilled workers is not enough to overcome rework levels. This scenario is similar to the effect of improving productivity 15%. Outsource labor is a good method handle short peaks in workload if the labor is highly skilled. But, it does not help in changing the ability to meet cost and schedule goals when the drivers of cost and schedule growth are caused by misses of 20% in planned productivity and 15% in planned quality levels.

6.2.7 Scenario G - Baseline Project (Imperfections and Instability included) with lower initial utilization

The last policy to be analyzed is staffing programs at higher levels initially to improve the ability to handle the quality and productivity problems as they arise. The added staffing provides added flexibility to the programs. The model has demonstrated that the introduction of problems drives higher cost in the programs as resource requirements grow to cover the additional work added by unintentional iterations. The problems in programs drive up overtime and increase fatigue, which further reduces productivity and quality levels. Staffing additions as the program nears completion add to the reduction in productivity and quality because new staff operates at lower productivity and quality and demands time for mentoring and training from higher skilled personnel. There is not enough time to acquire staff and train them to handle the tasks. In this analysis the model is configured to look at starting the programs with staffing at average utilization rates that are at or below 70% and checking to see if the programs modeled here can still complete the project without staffing increases and within schedule constraints. The reason for evaluating staffing at utilization levels below 70% is that the literature review revealed that waiting times significantly increase when resources are utilized at levels that are above 70%.

6.2.7.1 Staffing Utilization Set to 70% and to Level Required to Finish On Schedule

The first analysis run set the initial utilization rates of the staff at 70% instead assuming a full utilization level from the beginning. Since the first run did not meet any of the metrics, the second run determined staffing levels which met the level 1 schedule requirements. The tables below outline the metrics obtained during the analysis.

Project	Schedule (Months)	Schedule Slip (Months)	Delta % from Scenario A	Cost (Total Person*Months)	Cost Growth (Delta % from Scenario A)
Project 2	39.5	3.8	10.7%	6668	47.3%
Project 3	39.4375	3.4	9.5%	4804	39.1%
Project 4	42.875	4.1	10.6%	6543	41.2%
Total		3.8		18015	42.8%

Table 6-33 : Scenario G 70 Percent Cost and Schedule Differences from Baseline

Project	Schedule (Months)	Schedule Slip (Months)	Delta % from Scenario A	Cost (Total Person*Months)	Cost Growth (Delta % from Scenario A)
Project 2	35.875	0.2	0.5%	7661	69.2%
Project 3	36.3125	0.3	0.9%	5323	54.1%
Project 4	38.9375	0.2	0.5%	6930	49.5%
Total		0.2		19914	57.8%

Table 6-34 : Scenario G Cost and Schedule Differences from Baseline

The above tables demonstrate that for productivity and quality reductions from plan that are 20% and 15% lower than anticipated high levels of personnel are required to mitigate the risk to cost and schedule. Staffing at levels that utilize resources at a 70% level at the beginning of the program is not sufficient to mitigate the problems introduced that do not reflect the plan. The addition of personnel at the beginning of the program can reduce the schedule slip to only 0.2 months depending on the amount of personnel hired. The utilization rate to achieve this improvement is 40%-50% depending on the project. The drawback is that the entire amount of cost growth that scenario C introduces is built into the program from the beginning. Cost growth is only reduced from 64.5% to 57.8%. Since cost is measured in person*months to complete the project the cost growth is directly related to the growth in the initial amount of personnel required to complete the projects. To handle the high rework introduced by lower than anticipated quality cost growth is conceded at the very beginning of the program because of carrying costs of the personnel required to finish within schedule. The personnel required to complete the programs are shown in Figure 6-30. Scenario G does not require a ramp up in personnel to handle the added workload at the end of the project.

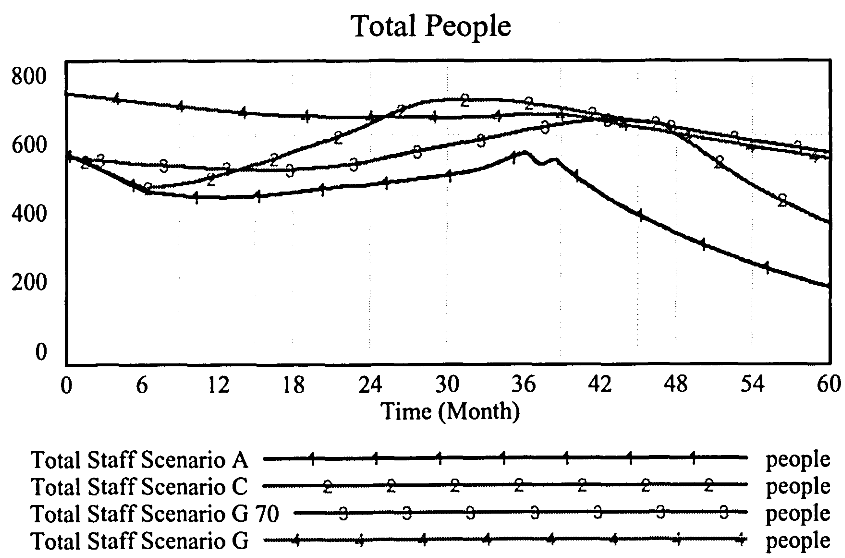

Figure 6-30: Total People required to complete all programs comparison

In these scenarios overtime is not required until the end of the program. The accumulated overtime for both cases is less than that required during Scenario A. Therefore productivity and quality do not suffer as much degradation as previous scenarios. At 70% initialization overtime is still significant at the end of the program since the amount of people carried by the program is not enough to handle the growth in the amount of work. Scenario G begins with utilization levels that are 40-50% depending on the project. Overtime use in this scenario is significantly less than any other scenario. Due to the reduced overtime fatigue is lowered and productivity and quality are raised. Productivity is shown to be almost equivalent to productivity levels of the scenario A plan in Figure 6-31.

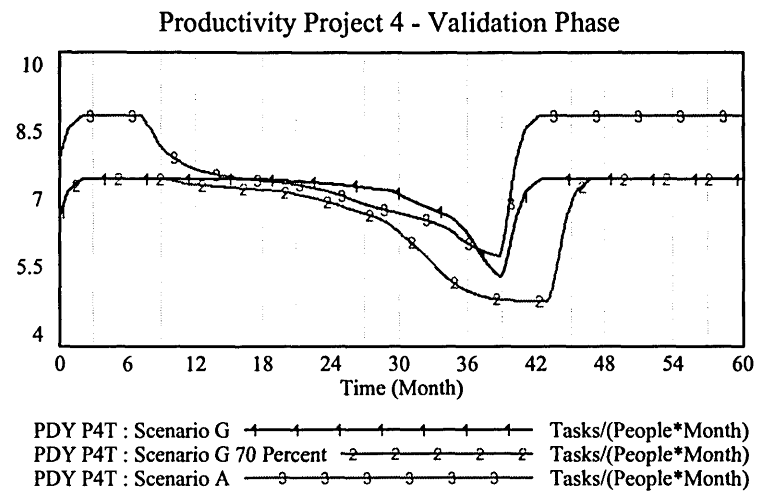

Figure 6-31 : Productivity for Scenario G Almost Equivalent to Scenario A Baseline

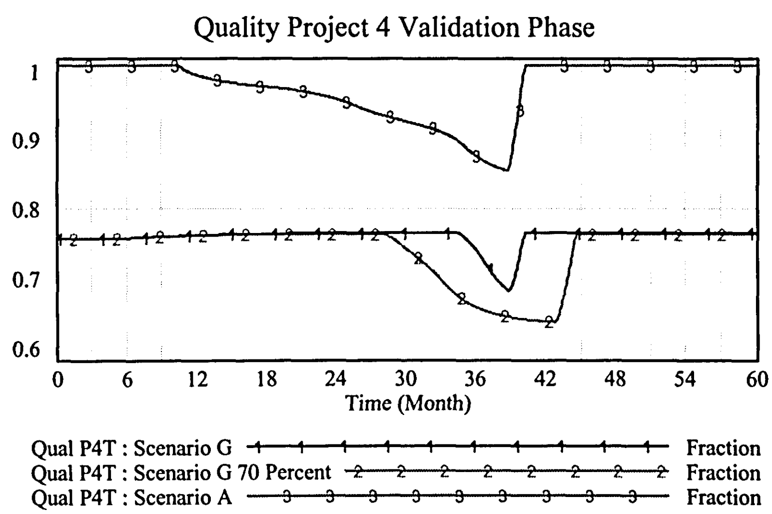

Figure 6-32 : Quality Levels are Still Significantly Below Perfect Execution

In this scenario low quality still generates much more rework than the baseline scenario A plan.

	Scenario G
Project	Increase in Rework in Validation Phase from Scenario A
Project 2	605%
Project 3	690%
Project 4	336%

Table 6-35 : Scenario G Increase in Rework

The additional staffing helps reduce fatigue and does not degrade performance, but lower than expected quality drives the project into requiring high levels of staff to complete the work. The cost over run is high relative to the baseline plan because such a high level of staffing is required to complete the program.

6.2.7.2 Summary

The policy of staffing a project with more personnel than initially required can reduce the schedule slip to zero, but in the case of reduced productivity levels of 15% and reduced quality levels of 20% cost growth will be 58%. If the project management decides to limit the amount of staffing so they are utilized at a 70% rate schedule will slip of 3.8 months with a cost growth of 43%. Programs that face a high degree of lower than expected productivity and quality have to significantly overstaff at the beginning of the project to handle the growth in work to do that occurs because of low quality. High initial staffing of projects eliminates the system dynamic effect of overtime causing fatigue and further reductions in quality and productivity. Employees are loaded to 50% of their capability at the beginning of the programs rather than 100% at the beginning of the program. As lower than expected quality begins to significantly add to the planned workload the added workforce can handle the growth in work. This policy trades cost to mitigate schedule risk. Projects that suffer high cost and schedule uncertainty without any other means of risk mitigation may have to overstaff to keep the ability to complete the projects on schedule.

6.3 Summary

The calibration of the model began with the assumption of a workload curve that was not stable due to timing of projects. Staffing levels were started at levels to cover projects that were completing the validation phase. Staffing levels in the Validation organization are high at completion of programs. The levels drop in the initial phases of a program due to business pressures to keep costs to a minimum, then the levels have to ramp up to handle high workloads as the validation phase begins late in the product development process.

In addition, projects are planned assuming that the programs will be completed with a minimum of problems. Quality levels are assumed to be near perfect. The amount of work that must be completed is assumed to be the work that is planned. If quality is lower than expected the assumption as to the amount of work that needs to be performed will be wrong. Also, productivity levels are assumed to be fairly high so that the work to be accomplished will be completed at a given rate. If productivity is lowered the program will fall behind schedule even without added work due to unintentional iterations. In the case of multi-project gas turbine product development programs the planned workload and productivity are optimistic. The cost and schedule goals are highly aggressive. The scenarios above analyze the plan, realistic outcomes of the projects, and various policies that can be utilized to mitigate the risk of overly optimistic and aggressive planned product development processes. The system dynamic model was calibrated to capture the realities of multi-project gas turbine product development processes.

The below Table summarizes the key metrics of the various scenarios explored.

Project	Schedule Metric	Cost Metric
Planned Project Completion (All Projects)	38 Months to Complete	Determined by SD Model
Scenario A (P&W Plan) Baseline Project - One project finishing, Three parallel projects starting.	38.75 Months to Complete	12616Person*Months
Scenario B Baseline Project with Imperfect Quality and Low Productivity	5.3 Months Slip Average	56.9% Cost Growth

Scenario C (P&W Reality) Baseline Project with Imperfections and Project Instability	7.0 Months Slip Average	64.6% Cost Growth
Scenario D Baseline Project (Imperfections and Instability included) with 15% Productivity Improvement	4.3 Months Slip Average	33.2% Cost Growth
Scenario D Baseline Project (Imperfections and Instability included) with Maximum Productivity Improvements in Learning Time	1.2 Months Slip Average	22.5% Cost Growth
Scenario D Baseline Project (Imperfections and Instability included) with Maximum Productivity Improvements	0.3 Months Slip Average	-26.5% Cost Growth
Scenario E Baseline Project (Imperfections and Instability included) with Quality Improved to Three Sigma	1.6 Months Slip Average	29.9% Cost Growth
Scenario E Baseline Project (Imperfections and Instability included) with Quality Improved to Six Sigma	0.4 Months Slip Average	21.0% Cost Growth
Scenario F Baseline Project (Imperfections and Instability included) utilizing highly skilled Outsource labor	2.7 Months Slip Average	24.6% Cost Growth
Scenario G Baseline Project (Imperfections and Instability included) with lower initial utilization of Resources	0.2 Months Slip Average	57.8% Cost Growth

The baseline case represents the perfect plan in a multi-project environment. The decisions about the amount of staffing required for the tasks are based on this perfect plan. As normal problems occur due to lower than planned productivity, poor quality, and project instabilities due to late discovery of quality issues overtime is used to keep projects on schedule. The model implemented 15% lower productivity and 20% lower than planned quality. A late problem that caused a 1.7 month schedule slip at the beginning of the validation phase of all projects were introduced into the model to simulate the realities of the gas turbine product development process. The introduction of these deviations from the planned performance increases overtime use and the resulting fatigue in the workforce. Productivity and quality are further degraded in the model due to the increase in fatigue. Once the available workforce is saturated in work overtime becomes ineffective in the ability to mitigate cost

and schedule risk. The only way to increase the rate at which work can be accomplished is to add staff. The new staff requires training and mentoring to produce high quality work. The new staff suffers from lower productivity and quality than the experienced staff. In addition the new staff drains productivity from the experienced staff on the program. The above mentioned system dynamic relationships result in an average 7-month schedule slip and a 64.6% cost overrun in the simulated projects. Overtime is utilized by the projects in the first three scenario to mitigate cost and schedule risk, but overtime causes fatigue and after 4 months of constant use is ineffective in mitigating the cost and schedule risk. The degraded quality and productivity of the workforce due to high levels of overtime use negates the added hours that overtime adds to the programs. The result in Scenario C demonstrates that cost and schedule growth is in evidence even though overtime is used. The results in Scenario C are representative of the amounts of growth seen within Pratt & Whitney during programs over the last 10 years. The risk framework showed that productivity and quality within the projects are key drivers of cost and schedule uncertainty. The company has implemented process improvements, new systems, and quality initiatives to improve productivity and quality within the program in response to the growth. Definition of processes can also have the effect of reducing training time of new staff and bringing quality and productivity levels up to more experienced employees faster. Programs are planned with a 15% improvement in productivity.
The analysis shows that improvements in productivity and quality can alleviate the system dynamic issues within the programs and improve cost and schedule metrics. The amount of the improvement is dependent on how much productivity improvement can realistically be achieved.
The model was tested with the assumed 15% improvement that programs are planned with. The model showed that a 15% improvement in productivity is not sufficient to alleviate a 20% lower than expected quality level. Schedule slips averaging 4.3 months and cost growth of 33.2% over the course of the programs were experienced with only a 15 % gain in productivity. Gains in

productivity on the order of 112% were required to mitigate cost and schedule risk within the programs. With over 80% gains in employee productivity and 80% gains in employee skill development times the schedule slip can be reduced to 0.3% and the cost will be 27% less than plan. Productivity gains of 80% are not impossible, but they have never been experienced over the full product development process. The high gains have only been experienced in select processes within the product development process. It is highly unlikely that productivity gains will be able to fully mitigate the problems seen with multi-project gas turbine engine product development processes Quality improvements were also highlighted in the risk framework for an ability to mitigate cost and schedule risk. Quality improvements can help reduce the rework and hence the workload within the program. Quality improvement programs such as Six Sigma and ACE at Pratt & Whitney attempt to drive base quality to extremely high levels.

The analysis showed that quality improvements to three sigma levels from 80% levels could reduce schedule growth to 1.6 months and cost growth to 29.9%. Additional improvements to quality that represent six sigma levels can mitigate the risk further to only 0.4 months schedule slip with a 21% cost growth. The improvements seen are significant, but as with productivity improvements the ability to achieve three sigma and six sigma quality levels across multi-project gas turbine product development processes are unlikely and costly. The cost of implementing these processes is not factored into the cost growth in this analysis.

Another method that can be employed to reduce time to bring employees up to higher quality and productivity levels faster is to use more experienced outsource help rather than hiring new hires to handle peak workloads. The analysis indicated that outsource labor can reduce the schedule growth to 2.7 months and the cost growth to 24.6% if outsource labor that is as skilled as expert employees can be found. The analysis assumed that a relatively high amount of outsource labor could be found. The improvements were a result of hiring 248 outsource labor personnel out of 642 total personnel in the program. The analysis showed that the sensitivity of hiring less skilled workers is significant. If labor is only as skilled as intermediate skill level staff the schedule growth is raised to 6.5 months with a cost growth of 50%. Very little risk to cost and schedule is mitigated in this scenario. Also, if the ability to hire labor is restricted the ability to mitigate cost and schedule risk is reduced. The analysis showed that by hiring a quarter less outsource resulted in an 80% growth in schedule slip. If the requirements for additional staffing are low enough that outsource help are available this method will help combat the vicious cycle of fatigue effects. At issue is the ability of outsource labor to fill the amount of labor required to keep the level 1 schedule. If the quality and productivity levels of the outsource labor are not equivalent of expert skilled staff, large amounts of outsource labor will still be required to complete the program on time.

Companies wish to keep costs to a minimum and this is the reason that staffing levels are kept to minimum levels and utilization factors are high for staff. On the other hand, process management techniques to reduce waiting time in projects and speed throughput rely on lower utilization of key resources. The analysis shows that if staffing levels are planned at a utilization factor that is 40%-50% for the work planned in multiple projects schedule slip can be reduced to an average of 0.2 months for the planned projects. The problem with the policy is that cost growth is 58% due to the higher cost of carrying the staff throughout the program. The lower utilization reduces the effect that as lower quality and productivity are introduced within the multi-project product development cycle the staff will be able to handle the added workload without the use of overtime. High overtime use introduces fatigue and lowers productivity and quality. From a business perspective high staffing levels are not a full program solution because the policy trades cost to mitigate schedule risk. The policy can be useful if there are many unknowns relative to quality and productivity in a project and schedule risk is paramount to cost risk.

7 Recommendations

Policy recommendations based on the findings of the analysis provide a more holistic method of applying the policies so that they better mitigate the risks to cost and schedule to complex, multi-project, aerospace product development processes and their relation to the scenarios presented in this thesis. The policies are summarized in the table below. Discussion of each policy follows.

Policy	Recommendations
Scenario A, B, and C Overtime **Overtime should be avoided for periods beyond the onset of fatigue. In the case of this model that period is 4 months.**	• Overtime should be kept to below the threshold fatigue begins to erode productivity and quality. • In the case of this model the time until the onset of fatigue is effectively 4 months, which is consistent with observations at this company. An additional month of overtime beyond the onset does not significantly degrade productivity and quality. • Once overtime reaches the duration that drives fatigue the ability to mitigate schedule risks by working extra hours can be degraded to the point that extra hours completely ineffective.
Scenario D Productivity Improvements **Process Improvements that cause 80% gains in employee productivity and skill advancement can mitigate Cost and Schedule Risk in programs with 80% Quality levels. Management Must Understand the Possible Process Improvement Levels and Monitor them during Program Execution.**	• 80% base productivity gains in the ability of staff to accomplish work along with 80% gains in time for staff to progress to the next skill level are enough to overcome a 20% reduction in planned quality. • Define average productivity and quality numbers for each Proficiency Level in Standard Work (every employee has a stated proficiency level, new, intermediate, expert, and supervisor) • Process must define tasks at finite enough level to establish productivity level. Standard work might have to be refined to better understand the volume of required iterations. • Establish the productivity level of staff during program to evaluate if productivity level is lower than planned • Assess capability to improve productivity to see if cost targets can be met.
Scenario E Quality Improvements **Improvements in Quality Levels from 80% Levels to Six Sigma levels Can Fully Mitigate Schedule Risk on Programs with 15% Lower than Planned Productivity. Management Must**	• Quality improvements of 15% can reduce a 7-month schedule slip by 5.5 months. • A 65% cost overrun will be reduced to 30% • Use turnback data to establish baseline rework levels and use to set base quality of programs.

<u>Understand the Current Quality Levels and the Possible Level of Improvement that can be achieved and Monitor them During Program Execution.</u>	• Establish gains that can be attained to evaluate if schedule can be mitigated with quality gains predicted.
Scenario F Outsourcing **<u>Outsource Labor is Limited in the Ability to Handle Peak Workloads in the Product Development Process. The Peak Workload must be Short in Duration and not Require Large Amounts of Outsource Labor to Mitigate Cost and Schedule Risk.</u>**	• Utilize outsourcing to reduce overtime if overtime levels are required beyond 4 months. • Establish the level of available, qualified outsource labor prior to project start • Skill level of outsource labor must be at least intermediate level
Scenario G Utilization Rate of Staffing **<u>Initially Staffing at 50%-70% Utilization Rates can Mitigate Schedule Risk on Programs with 15% Below Planned Productivity and 20% Below Planned Quality Levels. Use of this Policy is Limited to Programs with High Schedule Risk and Low Productivity and Quality with Low Probability of Improvement since Schedule Risk is Traded for Cost.</u>**	• Overstaffing program such that employees are utilized at 50% will mitigate schedule risk of 15% lower productivity than planned and 20% lower quality risk. • Cost overrun of 60% is given up at beginning of program. • Overstaffing program is useful when schedule risk is extremely high and cost risk is low. • Overstaffing can mitigate schedule risk when there is high uncertainty in the ability to hit planned productivity and quality numbers. • Initially staffing program with more people will mitigate schedule risk, but give up on cost. • Use standard work (task definition), productivity, and quality numbers to establish work to do and anticipated rework to "rightsize" overstaff to minimize cost.

Table 7-1 : Summary of Policy Recommendations

7.1 Use of Overtime

<u>Overtime should be avoided for periods beyond the onset of fatigue. In the case of this model that period is 4 months.</u>

Overtime use is one of the easiest management policies to implement that attempts to mitigate schedule risk. If the labor is unpaid for the extra hours worked, cost risk can also be mitigated. The analysis showed that overtime can be dangerous to use for extended periods of time. Overtime drives fatigue in each of the first three scenarios. Fatigue causes a system dynamic effect of reducing the quality and productivity of the employees in each of the scenarios tested. Reducing quality in the model adds to the tasks to complete the project by driving rework. Reducing productivity in the model slows the rate at which work is performed and further degrades the project performance. The model analysis indicates that for problems that are typically suffered in the product development process at Pratt & Whitney average schedule slips can be 7 months and cost overruns on the order of 64.6%. The intent of Scenario C was to define the real baseline performance of the multi-project product development process. To understand the contribution of overtime to the mitigation of cost and schedule risk Scenario C was run eliminating overtime. The schedule slip in this case was only an eighth of a month longer than 7 months. The negative effects of overtime over the multi-project product development process negate the extra work that the policy of overtime dictates. Employees suffer fatigue over the period that overtime is used and productivity and quality levels suffer. Low quality adds to the work that must be accomplished due to rework and low productivity affects the rate at which the work can be completed. For long term projects that require a 10 times more rework than originally planned overtime is ineffective in reducing cost and quality risk due to high levels of fatigue. Overtime can be used to mitigate cost and schedule risk but the staff must be monitored for the onset of fatigue.

7.2 Productivity Improvements

Process Improvements that cause 80% gains in employee productivity and skill advancement can mitigate Cost and Schedule Risk in programs with 80% Quality levels. Management Must Understand the Possible Process Improvement Levels and Monitor them during Program Execution.

The major thrust of Pratt & Whitney's management techniques to provide better products to the customer, faster than in prior product development processes, and at lower cost revolve around process improvement and quality initiatives. These practices are aimed at increasing productivity to maximum levels and raising quality levels to the highest possible standards. The problem with these initiatives is that realization of the full benefit of these practices to achieve high productivity and quality is extremely difficult to achieve in complex product development processes.

The system dynamic analysis shows that in the face of the inherent problems and instabilities in product development programs gains on the order of 80% are required in productivity to keep projects within cost and schedule constraints if quality levels are 20% lower than expected. In the case of this model the gains in productivity that can be made are in the amount of work that staff can complete in a given timeframe and in the time that the staff takes to advance from novice skill to intermediate to expert skill levels. The fact that most program planning is done assuming perfect execution so that costs and schedule can be minimized force improvements to productivity and quality to be higher than realistically achievable. The constraints cause the extended use of overtime, which has been demonstrated to be detrimental to the program schedule risk. Typical program management goals are productivity gains on the order of 15% within the programs. The model demonstrated that gains in productivity on the order of 15% are negated by extended use of overtime. A recent study conducted for Aviation Week and Space Technology by Boston based management consulting firm Pittiglio Rabin Todd & McGrath show that very few companies can achieve a high level of success employing Six Sigma techniques throughout the company.[35] Lean methods have enjoyed substantial success in manufacturing settings, but the transition to the product development setting have proven difficult. The promised timeframe to deliver the product to the customer often account for productivity gains not yet realized.

To reduce the risk to cost and schedule the project must understand baseline productivity within the program. Once the baseline productivity is understood realistic goals for improvement can be planned for in the product development process. As the process improvements enhance the productivity the project management will have better information with which to plan for resources

[35] Velocci, Anthony L., "Full Potential of Six Sigma Eludes Most Companies", Aviation Week and Space Technology, (September 30 2002)

during the program. Cost estimates for the remaining work will be more on target. Current efforts to define the tasks required to develop the product within the standard work process are the baseline efforts that can define productivity. Standard work in conjunction with data obtained in the EVMS process can be used to define the system performance. Productivity improvements can be measured by measuring the improvement in effort required to complete the tasks.

7.3 Quality Improvement

Improvements in Quality Levels from 80% Levels to Six Sigma levels Can Fully Mitigate Schedule Risk on Programs with 15% Lower than Planned Productivity. Management Must Understand the Current Quality Levels and the Possible Level of Improvement that can be achieved and Monitor them During Program Execution.

Quality initiatives such as Six Sigma and Pratt & Whitney's ACE process are intended to reduce defects to 3.4 defects in 1 million opportunities. Quality at this level reduces rework to virtually zero if policies such as overtime do not increase fatigue and reduce quality. The model demonstrated that the schedule slip related to lower than expected productivity and quality could be reduced to 0.4 months if the quality can be improved to six sigma levels. Cost growth is limited to 21%. The reason is that the lower than expected productivity drives slower progress and more person*months are required to complete the project.

The elimination of defects from processes is a key element of quality initiatives that are employed to reduce costs on programs in the aerospace industry. Defects in the process cause rework and adds to the work that must be done to accomplish goals of the projects. The systems dynamics model shows that rework can grow ten times the expected levels if quality is 20% lower than planned. The rework will drive high overtime levels because management uses the policy to mitigate cost and schedule risk within the programs. In addition the added rework drives the requirement to add new resources to complete the project. Hiring new staff drives the dynamic of low work quality of the added staff and the reduction of the existing staff's productivity due to mentoring and training of the new staff. The system dynamic effects work against efforts to improve quality levels in the process.

The methodology of the quality improvement programs involves techniques to define the problem and identification of the key drivers of turnbacks or reworks. The goal of quality improvement programs is to reduce the defects to 3.4 per 1 million opportunities or the six sigma level. The problem for the product development process is that without a highly defined process that identifies the number of tasks required to develop a gas turbine engine there is no way to figure out how many opportunities there is for defects to occur. Therefore there is no way to figure out what base the base quality level is within the program.

A method to determine the base quality level in the product development process will determine what effect quality improvements within the processes can have in mitigating risk to cost and schedule. The current practices of planning for perfect quality causes high cost and schedule risk since current quality levels are low based on the amount of rework. If initial quality is on the order of three sigma levels the schedule slip is still 1.6 months over the baseline plan. Program planning must take into account the expected number of turnbacks to properly plan project resources.

In addition there must be a realistic expectation during the life of the program on the amount of quality improvements that can be accomplished. A better definition of quality levels will reduce the cost uncertainty and enable managers to plan for enough resources to maintain schedules

7.4 Outsourcing

Outsource Labor is Limited in the Ability to Handle Peak Workloads in the Product Development Process. The Peak Workload must be Short in Duration and not Require Large Amounts of Outsource Labor to Mitigate Cost and Schedule Risk.

Outsource labor is a method that companies use to handle spikes in the resource requirements when instabilities and problems occur during programs that overtime cannot compensate for. To keep resource requirement oscillations from causing hiring and firing cycles within the permanent workforce management attempts to hire outsource personnel during periods of high need. The policy keeps the permanent workforce sized to handle the minimum workload. The multi-project system dynamic model demonstrated that planning for perfection while ignoring normal problems and instabilities causes the resource requirements to grow towards the end of the projects as rework adds to the work to do and the promised completion date nears. The model was set up with resource requirement oscillations that could be as high as 50% from valley to peak.

The use of outsource labor can reduce the schedule variance to 2.7 months and cost growth to 24.6%. With 80% of planned quality levels and 85% of planned productivity levels the additional staffing requirements become relatively large compared to the original requirements to achieve the above levels. Outsource labor requirements were 248 out of 642 people required in the program when expert skill level outsource labor were hired. The model shows that if the numbers of outsource personnel with the right skills can be found the advantage in training time, productivity, and quality that experienced outsource labor can bring can help reduce the cost and schedule uncertainty. But finding the right skill mix for the large number of outsource personnel that are required at the end of a complex product development process is not a reasonable assumption. The company must understand the capability of the outsource labor pool to determine the ability to handle peak workloads. The capability is determined by amount available and skill level of the outsource labor. Management should work with the outsource companies and gain an understanding of the amount of personnel available that can operate at a high level of productivity with high quality with minimum training. Utilization of high numbers of personnel who are at the same level as new hires add to the vicious cycle that is set up when training and mentoring reduce the productivity of experienced workers. The low skilled new employees are performing at a low productivity and quality levels. Outsourcing should be understood relative to the ability to mitigate schedule risk before being employed. Management should have a good understanding as to the amount and skill level of qualified labor that is available to better plan the ability to react to problems and instabilities.

7.5 Staffing Requirements

Initially Staffing at 50%-70% Utilization Rates can Mitigate Schedule Risk on Programs with 15% Below Planned Productivity and 20% Below Planned Quality Levels. Use of this Policy is Limited to Programs with High Schedule Risk and Low Productivity and Quality with Low Probability of Improvement since Schedule Risk is Traded for Cost.

The evaluation of the current process revealed that adequacy of staffing within the projects can reduce schedule uncertainty. Inadequate staffing requires overtime to be used at for extended periods. The multi-project system dynamic model demonstrated that extended use of overtime causes degraded productivity and quality that leads to poor progress and high rework. Cost and Schedule metrics are not held in this situation. To mitigate the risk to cost and schedule in production processes, resource utilization levels of 70% or below have been used because the low utilization has been shown to reduce waiting times and improve throughput. The analysis of the system dynamic model revealed that for the modeled product development processes with 70% initial utilization of resources a schedule slip of 3.6 months would still occur. To fully mitigate the schedule risk the initial utilization of the staffing resources on the three projects needs to be 40% to 50%. Keeping staff levels high at the beginning of projects causes high costs. The overall cost of the programs is 58% over the planned levels for projects that begin the product development process at 100% utilization factors.

Research in the areas of process management reveal that the product development process is in many ways similar to the manufacturing operations and application of lean manufacturing lessons may help reduce product development cycle time. In a multi-project environment that is organized with a matrix structure such as the one described in this paper resources are shared among programs. The system dynamic model shows that if resources are inadequate added rework can have a significant effect on completion time of the project. In addition, resources are stolen from less attractive programs to keep higher priority programs on schedule.

Planning for perfect execution of programs with minimum staffing levels that match low initial program workload and starting projects with almost full utilization of employees can cause substantial lengthening of the development times. In the last scenario the thesis used the system dynamics model to evaluate the effect of beginning programs at utilization levels below the 70% threshold. If quality and productivity levels are lower than planned and late problems occur in the development process the lower utilization rates allow the resources to be more flexible and more able to handle the added workload without suffering from detrimental system dynamic effects. Utilization rates of 40%-50% at the onset of the program were shown to be able to mitigate a 7-month schedule slip due to 15% lower than planned productivity and 20% lower than planned quality. The problem with this policy is that a 58% cost overrun will be built into the program from the beginning. Schedule risk in this policy is traded for program cost.

The amount of overstaffing required is dependent on the level that productivity and quality is below planned levels. The analysis points to the idea that programs must have a good idea of the baseline productivity and quality metrics before the lower initial utilization can mitigate the cost and schedule risks. If productivity and quality levels are understood the amount of required staffing will be able to be determined at a more finite level. Use of overstaffing can mitigate schedule risk if maintaining schedule is the most important factor in program success and productivity and quality levels are poorly understood and the expectation is that they will be below the planned levels.

7.6 Summary

Management of multi-project product development processes that have instabilities built into the system requires the ability to understand the planned iterations and the productivity and quality levels the workforce is able to achieve. Management typically plans for near flawless execution of programs and believes that the use of overtime will mitigate the schedule risks. The analysis showed that fatigue due to high overtime use will nullify the advantage of extracting more work from employees per week. The policy changes required during the management of programs must evaluate the system from a more holistic sense.

Overtime must be restricted to less than 4 months on a continual basis or fatigue levels will begin to add schedule slips back into the programs due to reduced productivity and quality. If overtime begins to accumulate beyond 4 months the use of outsource labor can quickly add to the workforce and reduce the levels of required overtime. Outsource labor skill level must be at least at an intermediate level to reduce overtime without adding rework and lowering productivity. There is no cost and schedule mitigation if the outsource labor is below intermediate skill level. If the outsource labor is of expert skill level the cost and schedule may be improved by the amount their average skill level is above an intermediate level. Programs must understand the quality of the outsource labor and the size of the labor pool to understand the ability of this policy to mitigate cost and schedule risk.

The above two risk mitigation policies are responses to unplanned work that stretches resources beyond their ability respond within normal work hours. The thesis also looked at the ability of productivity gains and quality gains to reduce the cost and schedule risk. Productivity gains are the most powerful tool to reduce cost and schedule overruns since the possibility exists to improve productivity beyond planned levels. Productivity gains on the order of 80% can fully mitigate 20% lower quality in the program. Productivity gains such as these are difficult to realize within a program, but data collected in lean initiatives indicate they are not unachievable. Baseline productivity levels should be understood. Productivity gains should be pursued aggressively to mitigate cost and schedule risk

Quality improvements usually are mentioned along with productivity improvements. The model results indicate that 20% gains in quality can mitigate 15% lower productivity than planned. The quality improvements reduce rework levels from 8 times to only double the planned amount. Even with lower than planned productivity the use of overtime does not significantly increase rework and lower productivity so that schedule slips occur due to overtime.

The two above policies indicate that productivity and quality are key metrics that should be fully understood. Planned levels of productivity and quality should be determined and metrics should be measured to see how real levels compare with planned levels. The metrics will indicate which policy will be most effective at relieving the pressure put on cost and schedule.

The last policy involving resource utilization is useful when the quality and productivity are not well defined and the expectation is that they will be significantly lower than planned. In addition schedule uncertainty should be a major concern for the program. Cost risk should not be a major factor to program success. Resource utilization within the programs is normally 100% and the use of overtime will be used to mitigate any schedule risk. When schedule uncertainty can not be tolerated and productivity and quality levels have a high probability of being 15-20% lower than targets staffing programs with double the planned number of personnel can reduce schedule slippage to zero. The cost of the program will suffer a 60% overrun, but the decision has to be that schedule must be met regardless of the cost.

The summary indicates that there are different policies that can be put into place in multi-project product development processes that can maintain cost and schedule requirements depending on expected levels of productivity and quality. The amount that productivity and quality are expected to under run the planned levels determines the best response. If the quality and productivity are expected to be close to levels that the program cost and schedule metrics are based upon overtime and outsource usage can be employed to mitigate cost and schedule risk. If the expectation that productivity and quality levels may be missed by 15-20% cost and productivity improvements must be aggressively pursued. When expectations are that the productivity and quality will be 15-20% lower than planned and the risk is high that they can not be significantly improved low utilization of staff may be employed in addition to aggressively pursuing productivity and quality improvements. This policy is most effective if schedule must be held at all costs. Staffing at levels that utilize the employee's time at a 50-70% level relative to the planned workload can reduce schedule uncertainty to near zero, but cost overruns will be approximately 60%.

The above policies provide methods to mitigate cost and schedule risk on multi-project product development programs that have inherent instabilities built into the programs.

8 Future Work

The work presented here attempts to show how a holistic approach toward managing the product development processes in the face of industry realities. The policy of not taking on projects or delaying them is not an option. The thesis gives overall policies that used together may provide a means of managing the complex interdependencies of multi-project product development process. The policies are general in scope. There needs to be more a more detailed analysis of the processes to determine if tasks can be more finitely defined in these complex processes.

Productivity and quality improvements are the key metric to determining if cost and schedule commitments can be met. Measurements that provide the base productivity of the workers need to be defined. Defining productivity involves figuring out the length of time one should be able to complete a task and the length of time it actually takes to perform a task. The ratio defined is a measure of process efficiency. Defining quality involves measuring turnbacks, but also knowledge of how many tasks are required to be completed. The tools are beginning to be employed that make gathering the data about product development tasks possible. SAP in conjunction with the EVMS process provides a measure of the hours that tasks take to complete. The ACE process utilizes databased decisions on what problems within the process need to have focused improvement efforts. Once the problems are identified kaizen type activities are utilized to improve the processes. A measure of the number of iterations that are required to complete the task successfully is known as process yield.[36] The yield and efficiency numbers determines the degree that the processes are in control. Utilization number for the groups responsible for the completing tasks within the product development process can also be determined by gathering data on the hours that each resource is used to complete tasks. Typically the technicians and mechanics will not be fully utilized, but the engineers are. Determination of utilization numbers for the groups may provide insight into possible bottlenecks.

The definition of all tasks has been determined with standard work. The difficulty is determining how to measure task lengths and separate waiting time from the task length. Most tasks in the product development process do not start and stop sequentially. Tasks are started, then they wait for other data to continue, and then the tasks are restarted. This cycle may occur many times. In addition the process my iterate back to the beginning multiple times before completion. The data must be gathered to analyze where improvements must be targeted. Data can be gathered using the WBS and EVMS processes to determine the amount of manpower used on tasks. Measurements of productivity and quality gains can be made throughout the process. Realistic improvements can be planned for within the programs based on the history and estimated gains new processes or tools may add. The mitigation plans can be implemented based on a real plan of the ability of the program to meet expected productivity and quality levels. At this time the improvements are assumed. The plan is based on assumptions and not based on real data on productivity and quality.

An example of how the detail in unison with the results here would work would be as follows. The measurement of task length would determine that productivity and quality were 30% below plan and history had shown 5% improvements year over year. Productivity and quality gains could not be counted on in this case to mitigate cost and schedule risk based on metrics. Overtime and could not cover the increase in workload without significant use. The resulting fatigue would negate the effects. Outsourcing could not provide enough resources to mitigate the risk. The option would be to overstaff the program and admit that cost was going to be higher than expected. Otherwise

---

[36] Adler, Paul, Mandelbaum, Avi, Nguyen, Vien, and Schwerer, Elizabeth, "Getting the Most our of Your Product Development Process", Harvard Business Review, (March-April 1996)

specific productivity and quality improvements would have to be made and the effect of the improvements would have to be determined if a different mitigation strategy could be used.

Appendix -A

Simulation Data

Model Constant Spreadsheets

Scenario A

Project	Priority	Starting NR	Starting IR	Starting ER	Starting NH	Starting IH	Starting EH	Starting ND	Starting ID	Starting ED	Starting NT	Starting IT	Starting ET
P1	1	0.00	50.00	19.00	0.00	50.00	31.00	0.00	50.00	41.00	0.00	125.00	140.00
P2	3	0.00	1.00	1.00	2.00	1.00	1.00	2.00	1.00	1.00	3.00	1.00	1.00
P3	3	0.00	1.00	1.00	1.00	1.00	1.00	2.00	1.00	1.00	3.00	1.00	1.00
P4	3	0.00	1.00	1.00	2.00	1.00	1.00	2.00	1.00	1.00	3.00	1.00	1.00
			Initial Concept Staff (people)			Initial Preliminary Design Staff (people)			Initial Detail Design Staff (people)			Initial Validation Staff (people)	

WTD R	WTD H	WTD D	WTD T	T R	T H	T D	T R	TTS	Complexity	PD Prob Pulse Start	PD Prob Pulse Work	DD Prob Pulse Start	DD Prob Pulse Work	Valid. Prob Pulse Start	Valid. Prob Pulse Work	Project
10	10	10	10	6	6	6	38	0	1	0	0	0	0	0	0	P1
1600	5600	11700	50400	6	12	27	35.5	0	1	0	0	0	0	0	0	P2
1000	3500	7540	42000	8	15	22	36	2	1	0	0	0	0	0	0	P3
2000	6000	11800	47880	12	18	26	38	6	1	0	0	0	0	0	0	P4
	Initial Work To Do (lines)				Initial Duration (months)											

Attractiveness Weights

Priority	2
Bug Ratio	1
Staffing Gap	1
Complexity	1

People Movement Times

TimeToMoveIn	2	Project 1
TimeToMoveOut	12	2
TimeToHire	8	
TimeToDownSize	24	1
TimeForAttrition	12	

	Requirements	HLD	Development	Test
NoviceToIntermediateTime	24	24	24	24
IntermediateToExpertTime	24	24	24	24
MinimumRemainingTime	2	1	0.5	0.25
TimeToGetFatigued	3	3	3	3
TimeToPerceivePDY	2	2	2	2
MaximumQuality	1	1	1	1
NormalProductivity	10	10	10	7
BugFindTime	2	2	1.5	0.5
MaximumStaff	90	110	143	537

People Factors

NoviceMultiplier	1
IntMultiplier	1.5
ExpertMultiplier	3
NoviceSkillEffectOnQuality	1
IntSkillEffectOnQuality	1
ExpertSkillEffectOnQuality	1

Scenario B

Project	Priority	Starting NR	Starting IR	Starting ER	Starting NH	Starting IH	Starting EH	Starting ND	Starting ID	Starting ED	Starting NT	Starting IT	Starting ET
P1	1	0.00	50.00	19.00	0.00	50.00	31.00	0.00	50.00	41.00	0.00	125.00	140.00
P2	3	0.00	1.00	1.00	2.00	1.00	1.00	2.00	1.00	1.00	3.00	1.00	1.00
P3	3	0.00	1.00	1.00	1.00	1.00	1.00	2.00	1.00	1.00	3.00	1.00	1.00
P4	3	0.00	1.00	1.00	2.00	1.00	1.00	2.00	1.00	1.00	3.00	1.00	1.00
			Initial Concept Staff (people)			Initial Preliminary Design Staff (people)			Initial Detail Design Staff (people)			Initial Validation Staff (people)	

WTD R	WTD H	WTD D	WTD T	T R	T H	T D	T R	TTS	Complexity	PD Prob Pulse Start	PD Prob Pulse Work	DD Prob Pulse Start	DD Prob Pulse Work	Valid. Prob Pulse Start	Valid. Prob Pulse Work	Project
10	10	10	600	3	3	3	38	0	1	0	0	0	0	0	0	P1
1600	5600	11700	50400	6	12	27	35.5	0	10	0	0	0	0	0	0	P2
1000	3500	7540	42000	8	15	22	36	2	10	0	0	0	0	0	0	P3
2000	6000	11800	47880	12	18	26	38	6	10	0	0	0	0	0	0	P4
	Initial Work To Do (Tasks)				Initial Duration (months)											

Attractiveness Weights

Priority	2
Bug Ratio	1
Staffing Gap	1
Complexity	1

People Movement Times

TimeToMoveIn	2	Project 1
TimeToMoveOut	12	2
TimeToHire	8	
TimeToDownSize	24	1
TimeForAttrition	12	

People Factors

NoviceMultiplier	1
IntMultiplier	1.5
ExpertMultiplier	3
NoviceSkillEffectOnQuality	0.92
IntSkillEffectOnQuality	0.95
ExpertSkillEffectOnQuality	0.98

	Requirements	**HLD**	**Development**	**Test**
NoviceToIntermediateTime	24	24	24	24
IntermediateToExpertTime	24	24	24	24
MinimumRemainingTime	2	1	0.5	0.25
TimeToGetFatigued	3	3	3	3
TimeToPerceivePDY	2	2	2	2
MaximumQuality	0.92	0.94	0.96	0.98
NormalProductivity	10	10	10	7
BugFindTime	2	2	1.5	0.5
MaximumStaff	90	110	143	537

Scenario C

Project	Priority	Starting NR	Starting IR	Starting ER	Starting NH	Starting IH	Starting EH	Starting ND	Starting ID	Starting ED	Starting NT	Starting IT	Starting ET
P1	1	0.00	50.00	19.00	0.00	50.00	31.00	0.00	50.00	41.00	0.00	125.00	140.00
P2	3	0.00	1.00	1.00	2.00	1.00	1.00	2.00	1.00	1.00	3.00	1.00	1.00
P3	3	0.00	1.00	1.00	1.00	1.00	1.00	2.00	1.00	1.00	3.00	1.00	1.00
P4	3	0.00	1.00	1.00	2.00	1.00	1.00	2.00	1.00	1.00	3.00	1.00	1.00
		Initial Concept Staff (people)			Initial Preliminary Design Staff (people)			Initial Detail Design Staff (people)			Initial Validation Staff (people)		

WTD R	WTD H	WTD D	WTD T	T R	T H	T D	T R	TTS	Complexity	PD Prob Pulse Start	PD Prob Pulse Work	DD Prob Pulse Start	DD Prob Pulse Work	Valid. Prob Pulse Start	Valid. Prob Pulse Work	Project
10	10	10	600	3	3	3	38	0	1	0	0	0	0	0	0	P1
1600	5600	11700	50400	6	12	27	35.5	0	10	0	0	0	0	30	2400	P2
1000	3500	7540	42000	8	15	22	36	2	10	0	0	0	0	25	2000	P3
2000	6000	11800	47880	12	18	26	38	6	10	0	0	0	0	30	2000	P4
Initial Work To Do (tasks)				Initial Duration (months)												

Attractiveness Weights

Priority	2
Bug Ratio	1
Staffing Gap	1
Complexity	1

People Movement Times

TimeToMoveIn	2	Project 1
TimeToMoveOut	12	2
TimeToHire	8	
TimeToDownSize	24	1
TimeForAttrition	12	

People Factors

NoviceMultiplier	1
IntMultiplier	1.5
ExpertMultiplier	3
NoviceSkillEffectOnQuality	0.92
IntSkillEffectOnQuality	0.95
ExpertSkillEffectOnQuality	0.98

	Requirements	HLD	Development	Test
NoviceToIntermediateTime	24	24	24	24
IntermediateToExpertTime	24	24	24	24
MinimumRemainingTime	2	1	0.5	0.25
TimeToGetFatigued	3	3	3	3
TimeToPerceivePDY	2	2	2	2
MaximumQuality	0.92	0.94	0.96	0.98
NormalProductivity	10	10	10	7
BugFindTime	2	2	1.5	0.5
MaximumStaff	90	110	143	537

Scenario D

Project	Priority	Starting NR	Starting IR	Starting ER	Starting NH	Starting IH	Starting EH	Starting ND	Starting ID	Starting ED	Starting NT	Starting IT	Starting ET
P1	1	0.00	50.00	19.00	0.00	50.00	31.00	0.00	50.00	41.00	0.00	125.00	140.00
P2	3	0.00	1.00	1.00	2.00	1.00	1.00	2.00	1.00	1.00	3.00	1.00	1.00
P3	3	0.00	1.00	1.00	1.00	1.00	1.00	2.00	1.00	1.00	3.00	1.00	1.00
P4	3	0.00	1.00	1.00	2.00	1.00	1.00	2.00	1.00	1.00	3.00	1.00	1.00
		Initial Concept Staff (people)			Initial Preliminary Design Staff (people)			Initial Detail Design Staff (people)			Initial Validation Staff (people)		

WTD R	WTD H	WTD D	WTD T	T R	T H	T D	T R	TTS	Complexity	PD Prob Pulse Start	PD Prob Pulse Work	DD Prob Pulse Start	DD Prob Pulse Work	Valid. Prob Pulse Start	Valid. Prob Pulse Work	Project
10	10	10	600	3	3	3	38	0	1	0	0	0	0	0	0	P1
1600	5600	11700	50400	6	12	27	35.5	0	10	0	0	0	0	30	2400	P2
1000	3500	7540	42000	8	15	22	36	2	10	0	0	0	0	25	2000	P3
2000	6000	11800	47880	12	18	26	38	6	10	0	0	0	0	30	2000	P4
Initial Work To Do (tasks)				Initial Duration (months)												

Attractiveness Weights

Priority	2
Bug Ratio	1
Staffing Gap	1
Complexity	1

People Movement Times

TimeToMoveIn	2	Project 1
TimeToMoveOut	12	2
TimeToHire	8	
TimeToDownSize	24	1
TimeForAttrition	12	

	Requirements	HLD	Development	Test
NoviceToIntermediateTime	4.8	4.8	4.8	4.8
IntermediateToExpertTime	4.8	4.8	4.8	4.8
MinimumRemainingTime	2	1	0.5	0.25
TimeToGetFatigued	3	3	3	3
TimeToPerceivePDY	2	2	2	2
MaximumQuality	0.92	0.94	0.96	0.98
NormalProductivity	10	10	10	12.6
BugFindTime	2	2	1.5	0.5
MaximumStaff	90	110	143	537

People Factors

NoviceMultiplier	1
IntMultiplier	1.5
ExpertMultiplier	3
NoviceSkillEffectOnQuality	0.92
IntSkillEffectOnQuality	0.95
ExpertSkillEffectOnQuality	0.98

<u>Scenario E</u>

Project	Priority	Starting NR	Starting IR	Starting ER	Starting NH	Starting IH	Starting EH	Starting ND	Starting ID	Starting ED	Starting NT	Starting IT	Starting ET
P1	1	0.00	50.00	19.00	0.00	50.00	31.00	0.00	50.00	41.00	0.00	125.00	140.00
P2	3	0.00	1.00	1.00	2.00	1.00	1.00	2.00	1.00	1.00	3.00	1.00	1.00
P3	3	0.00	1.00	1.00	1.00	1.00	1.00	2.00	1.00	1.00	3.00	1.00	1.00
P4	3	0.00	1.00	1.00	2.00	1.00	1.00	2.00	1.00	1.00	3.00	1.00	1.00
			Initial Concept Staff (people)			Initial Preliminary Design Staff (people)			Initial Detail Design Staff (people)			Initial Validation Staff (people)	

WTD R	WTD H	WTD D	WTD T	T R	T H	T D	T R	TTS	Complexity	PD Prob Pulse Start	PD Prob Pulse Work	DD Prob Pulse Start	DD Prob Pulse Work	Valid. Prob Pulse Start	Valid. Prob Pulse Work	Project
10	10	10	600	3	3	3	38	0	1	0	0	0	0	0	0	P1
1600	5600	11700	50400	6	12	27	35.5	0	10	0	0	0	0	30	2400	P2
1000	3500	7540	42000	8	15	22	36	2	10	0	0	0	0	25	2000	P3
2000	6000	11800	47880	12	18	26	38	6	10	0	0	0	0	30	2000	P4
	Initial Work To Do (tasks)				Initial Duration (months)											

Attractiveness Weights

Priority	2
Bug Ratio	1
Staffing Gap	1
Complexity	1

People Movement Times

TimeToMoveIn	2	Project 1
TimeToMoveOut	12	2
TimeToHire	8	
TimeToDownSize	24	1
TimeForAttrition	12	

People Factors

NoviceMultiplier	1
IntMultiplier	1.5
ExpertMultiplier	3
NoviceSkillEffectOnQuality	0.99
IntSkillEffectOnQuality	0.99
ExpertSkillEffectOnQuality	0.99

	Requirements	HLD	Development	Test
NoviceToIntermediateTime	24	24	24	24
IntermediateToExpertTime	24	24	24	24
MinimumRemainingTime	2	1	0.5	0.25
TimeToGetFatigued	3	3	3	3
TimeToPerceivePDY	2	2	2	2
MaximumQuality	0.99	0.99	0.99	0.99
NormalProductivity	10	10	10	7
BugFindTime	2	2	1.5	0.5
MaximumStaff	90	110	143	537

Scenario F

Project	Priority	Starting NR	Starting IR	Starting ER	Starting NH	Starting IH	Starting EH	Starting ND	Starting ID	Starting ED	Starting NT	Starting IT	Starting ET
P1	1	0.00	50.00	19.00	0.00	50.00	31.00	0.00	50.00	41.00	0.00	125.00	140.00
P2	3	0.00	1.00	1.00	2.00	1.00	1.00	2.00	1.00	1.00	3.00	1.00	1.00
P3	3	0.00	1.00	1.00	1.00	1.00	1.00	2.00	1.00	1.00	3.00	1.00	1.00
P4	3	0.00	1.00	1.00	2.00	1.00	1.00	2.00	1.00	1.00	3.00	1.00	1.00
			Initial Concept Staff (people)			Initial Preliminary Design Staff (people)			Initial Detail Design Staff (people)			Initial Validation Staff (people)	

WTD R	WTD H	WTD D	WTD T	T R	T H	T D	T R	TTS	Complexity	PD Prob Pulse Start	PD Prob Pulse Work	DD Prob Pulse Start	DD Prob Pulse Work	Valid. Prob Pulse Start	Valid. Prob Pulse Work	Project
10	10	10	600	3	3	3	38	0	1	0	0	0	0	0	0	P1
1600	5600	11700	50400	6	12	27	35.5	0	10	0	0	0	0	30	2400	P2
1000	3500	7540	42000	8	15	22	36	2	10	0	0	0	0	25	2000	P3
2000	6000	11800	47880	12	18	26	38	6	10	0	0	0	0	30	2000	P4
	Initial Work To Do (tasks)				Initial Duration (months)											

Attractiveness Weights

Priority	2
Bug Ratio	1
Staffing Gap	1
Complexity	1

People Movement Times

TimeToMoveIn	2	Project 1
TimeToMoveOut	12	2
TimeToHire	2	
TimeToDownSize	1000	1000
TimeForAttrition	12	

People Factors

NoviceMultiplier	3
IntMultiplier	1.5
ExpertMultiplier	3
NoviceSkillEffectOnQuality	0.98
IntSkillEffectOnQuality	0.95
ExpertSkillEffectOnQuality	0.98

	Requirements	HLD	Development	Test
NoviceToIntermediateTime	1000	1000	1000	1000
IntermediateToExpertTime	24	24	24	24
MinimumRemainingTime	2	1	0.5	0.25
TimeToGetFatigued	3	3	3	3
TimeToPerceivePDY	2	2	2	2
MaximumQuality	0.92	0.94	0.96	0.98
NormalProductivity	10	10	10	7
BugFindTime	2	2	1.5	0.5
MaximumStaff	90	110	143	400

Scenario G

Project	Priority	Starting NR	Starting IR	Starting ER	Starting NH	Starting IH	Starting EH	Starting ND	Starting ID	Starting ED	Starting NT	Starting IT	Starting ET
P1	1	0.00	20.00	19.00	0.00	60.00	31.00	0.00	50.00	41.00	0.00	1.00	1.00
P2	3	0.00	1.00	1.00	2.00	1.00	1.00	2.00	1.00	1.00	3.00	48.00	48.00
P3	3	0.00	1.00	1.00	1.00	1.00	1.00	2.00	1.00	1.00	3.00	42.00	42.00
P4	3	0.00	1.00	1.00	2.00	1.00	1.00	2.00	1.00	1.00	3.00	51.00	51.00
			Initial Concept Staff (people)			Initial Preliminary Design Staff (people)			Initial Detail Design Staff (people)			Initial Validation Staff (people)	

WTD R	WTD H	WTD D	WTD T	T R	T H	T D	T R	TTS	Complexity	PD Prob Pulse Start	PD Prob Pulse Work	DD Prob Pulse Start	DD Prob Pulse Work	Valid. Prob Pulse Start	Valid. Prob Pulse Work	Project
10	10	10	600	3	3	3	38	0	1	0	0	0	0	0	0	P1
1600	5600	11700	50400	6	12	27	35.5	0	10	0	0	0	0	30	2400	P2
1000	3500	7540	42000	8	15	22	36	2	10	0	0	0	0	25	2000	P3
2000	6000	11800	47880	12	18	26	38	6	10	0	0	0	0	30	2000	P4
	Initial Work To Do (tasks)				Initial Duration (months)											

Attractiveness Weights

Priority	2
Bug Ratio	1
Staffing Gap	1
Complexity	1

People Movement Times

TimeToMoveIn	2	Project 1
TimeToMoveOut	12	2
TimeToHire	8	
TimeToDownSize	24	1
TimeForAttrition	12	

People Factors

NoviceMultiplier	1
IntMultiplier	1.5
ExpertMultiplier	3
NoviceSkillEffectOnQuality	0.92
IntSkillEffectOnQuality	0.95
ExpertSkillEffectOnQuality	0.98

	Requirements	HLD	Development	Test
NoviceToIntermediateTime	24	24	24	24
IntermediateToExpertTime	24	24	24	24
MinimumRemainingTime	2	1	0.5	0.25
TimeToGetFatigued	3	3	3	3
TimeToPerceivePDY	2	2	2	2
MaximumQuality	0.92	0.94	0.96	0.98
NormalProductivity	10	10	10	7
BugFindTime	2	2	1.5	0.5
MaximumStaff	90	110	143	537

Appendix -B

```
Total Work=
	WorkToDo P4R+WorkToDo P4H+WorkToDo P4D+WorkToDo P4T
	~	lines
	~		|

Attractiveness P3T=
	(Bug ratio effect on attractiveness P3T*Bug Ratio Weight+Priority effect on attractiveness P3T\
		*Priority Weight +Staffing Gap effect on attractiveness P3T*Staffing Gap Weight + Complexity effect on attractiveness P3\
		*Complexity Weight)*Active P3T
	~	dmnl
	~		|

WorkToDo P4T= INTEG (
	FindBugs P4T-Doing P4T+Test Start Rate P4T+P4T Problem Pulse,
		0)
	~	lines
	~		|

P4D Problem Initial WorkToDo=
	Get XLS Constants('ModelConstants.xls','Project Constants','ab5')
	~	lines
	~		|

P1IT Rate=
	if then else(GapP1IT>0,Min(GapP1IT, IT Control)/TimeToMoveIn,GapP1IT/TimeToMoveOutP1\
		)
	~	people/Month
	~		|

P3H Problem Initial WorkToDo=
	Get XLS Constants('ModelConstants.xls','Project Constants','z4')
	~	lines
	~		|

P3H Problem Pulse=
	P3H Problem Initial WorkToDo*pulse(P3H Time of Problem,TIME STEP)*(1/TIME STEP)
	~
	~		|

P4T Problem Initial WorkToDo=
	Get XLS Constants('ModelConstants.xls','Project Constants','ad5')
	~	lines
	~		|

P4T Problem Pulse=
	P4T Problem Initial WorkToDo*pulse(P4T Time of Problem,TIME STEP)*(1/TIME STEP)
	~	lines/Month
	~		|

P4T Time of Problem=
	Get XLS Constants('ModelConstants.xls','Project Constants','ac5')
	~	Month
	~		|

P1NH Rate=
	if then else(GapP1NH>0,Min(GapP1NH, NH Control)/TimeToMoveIn,GapP1NH/TimeToMoveOutP1\
		)
	~	people/Month
	~		|

P1ET Rate=
	if then else(GapP1ET>0,Min(GapP1ET, ET Control)/TimeToMoveIn,GapP1ET/TimeToMoveOutP1\
		)
	~	people/Month
	~		|

P3D Problem Initial WorkToDo=
	Get XLS Constants('ModelConstants.xls','Project Constants','ab4')
	~	lines
```

```
	~		|

P1NR Rate=
	if then else(GapP1NR>0,Min(GapP1NR, NR Control)/TimeToMoveIn,GapP1NR/TimeToMoveOutP1\
		)
	~	people/Month
	~		|

P3D Time of Problem=
	Get XLS Constants('ModelConstants.xls','Project Constants','aa4')
	~	Month
	~		|

P4D Problem Pulse=
	P4D Problem Initial WorkToDo*pulse(P4D Time of Problem,TIME STEP)*(1/TIME STEP)
	~	lines/Month
	~		|

P1NT Rate=
	if then else(GapP1NT>0,Min(GapP1NT, NT Control)/TimeToMoveIn,GapP1NT/TimeToMoveOutP1\
		)
	~	people/Month
	~		|

P4H Problem Initial WorkToDo=
	Get XLS Constants('ModelConstants.xls','Project Constants','z5')
	~	lines
	~		|

P4H Problem Pulse=
	P4H Problem Initial WorkToDo*pulse(P4H Time of Problem,TIME STEP)*(1/TIME STEP)
	~
	~		|

P4H Time of Problem=
	Get XLS Constants('ModelConstants.xls','Project Constants','y5')
	~	Month
	~		|

P2H Time of Problem=
	Get XLS Constants('ModelConstants.xls','Project Constants','y3')
	~	Month
	~		|

TimeToMoveOutP1=
	Get XLS Constants('ModelConstants.xls','Portfolio Constants','c10')
	~
	~		|

P1IR Rate=
	if then else(GapP1IR>0,Min(GapP1IR, IR Control)/TimeToMoveIn,GapP1IR/TimeToMoveOutP1\
		)
	~	people/Month
	~		|

WorkToDo P4H= INTEG (
	FindBugs P4H-Doing P4H+HLD Start Rate P4H+P4H Problem Pulse,
		0)
	~	lines
	~		|

P3D Problem Pulse=
	P3D Problem Initial WorkToDo*pulse(P3D Time of Problem,TIME STEP)*(1/TIME STEP)
	~
	~		|

WorkToDo P2H= INTEG (
	FindBugs P2H-Doing P2H+HLD Start Rate P2H+P2H Problem Pulse,
		0)
	~	lines
	~		|

P1ED Rate=
```

```
	if then else(GapP1ED>0,Min(GapP1ED, ED Control)/TimeToMoveIn,GapP1ED/TimeToMoveOutP1\
		)
	~	people/Month
	~		|

P1ID Rate=
	if then else(GapP1ID>0,Min(GapP1ID, ID Control)/TimeToMoveIn,GapP1ID/TimeToMoveOutP1\
		)
	~	people/Month
	~		|

P1EH Rate=
	if then else(GapP1EH>0,Min(GapP1EH, EH Control)/TimeToMoveIn,GapP1EH/TimeToMoveOutP1\
		)
	~	people/Month
	~		|

P2H Problem Initial WorkToDo=
	Get XLS Constants('ModelConstants.xls','Project Constants','z3')
	~	lines
	~		|

P1ER Rate=
	if then else(GapP1ER>0,Min(GapP1ER, ER Control)/TimeToMoveIn,GapP1ER/TimeToMoveOutP1\
		)
	~	people/Month
	~		|

P1IH Rate=
	if then else(GapP1IH>0,Min(GapP1IH, IH Control)/TimeToMoveIn,GapP1IH/TimeToMoveOutP1\
		)
	~	people/Month
	~		|

WorkToDo P4D= INTEG (
	FindBugs P4D-Doing P4D+Start Dev Rate P4D+P4D Problem Pulse,
		0)
	~	lines
	~		|

WorkToDo P3D= INTEG (
	FindBugs P3D-Doing P3D+Start Dev Rate P3D+P3D Problem Pulse,
		0)
	~	lines
	~		|

P3H Time of Problem=
	Get XLS Constants('ModelConstants.xls','Project Constants','y4')
	~	Month
	~		|

P2H Problem Pulse=
	P2H Problem Initial WorkToDo*pulse(P2H Time of Problem,TIME STEP)*(1/TIME STEP)
	~	lines/Month
	~		|

WorkToDo P3H= INTEG (
	FindBugs P3H-Doing P3H+HLD Start Rate P3H+P3H Problem Pulse,
		0)
	~	lines
	~		|

P4D Time of Problem=
	Get XLS Constants('ModelConstants.xls','Project Constants','aa5')
	~	Month
	~		|

P1ND Rate=
	if then else(GapP1ND>0,Min(GapP1ND, ND Control)/TimeToMoveIn,GapP1ND/TimeToMoveOutP1\
		)
	~	people/Month
	~		|
```

```
P3T Problem Pulse=
	P3T Problem Initial WorkToDo*pulse(P3T Time of Problem,TIME STEP)*(1/TIME STEP)
	~	lines/Month
	~		|

P3T Time of Problem=
	Get XLS Constants('ModelConstants.xls','Project Constants','ac4')
	~	Month
	~		|

WorkToDo P2T= INTEG (
	FindBugs P2T-Doing P2T+Test Start Rate P2T+P2T Problem Pulse,
		0)
	~	lines
	~		|

P2T Problem Pulse=
	P2T Problem Initial WorkToDo*pulse(P2T Time of Problem,TIME STEP)*(1/TIME STEP)
	~	lines/Month
	~		|

P2T Problem Initial WorkToDo=
	Get XLS Constants('ModelConstants.xls','Project Constants','ad3')
	~	lines
	~		|

WorkToDo P3T= INTEG (
	FindBugs P3T-Doing P3T+Test Start Rate P3T+P3T Problem Pulse,
		0)
	~	lines
	~		|

P2T Time of Problem=
	Get XLS Constants('ModelConstants.xls','Project Constants','ac3')
	~
	~		|

P3T Problem Initial WorkToDo=
	Get XLS Constants('ModelConstants.xls','Project Constants','ad4')
	~	lines
	~		|

P2D Time of Problem=
	Get XLS Constants('ModelConstants.xls','Project Constants','aa3')
	~	Month
	~		|

Downsize Rate NH=
	if then else(Downsize Goal NH > 0, Min(Downsize Goal NH, NH Control)/(Time to downsize\
		/Outsource Hire Normalizer),0)
	~	people/Month
	~		|

WorkToDo P2D= INTEG (
	FindBugs P2D-Doing P2D+Start Dev Rate P2D+P2D Problem Pulse,
		0)
	~	lines
	~		|

Downsize Rate NT=
	if then else(Downsize Goal NT > 0, Min(Downsize Goal NT, NT Control)/(Time to downsize\
		/Outsource Hire Normalizer),0)
	~	people/Month
	~		|

P2D Problem Initial WorkToDo=
	Get XLS Constants('ModelConstants.xls','Project Constants','ab3')
	~	lines
	~		|

Downsize Rate ND=
	if then else(Downsize Goal ND > 0, Min(Downsize Goal ND, ND Control)/(Time to downsize\
		/Outsource Hire Normalizer),0)
```

```
	~	people/Month
	~		|

P2D Problem Pulse=
	P2D Problem Initial WorkToDo*pulse(P2D Time of Problem,TIME STEP)*(1/TIME STEP)
	~	lines/Month
	~		|

Downsize Rate NR=
	if then else(Downsize Goal NR > 0, Min(Downsize Goal NR, NR Control)/(Time to downsize\
		/Outsource Hire Normalizer),0)
	~	people/Month
	~		|

Outsource Hire Normalizer=
	Get XLS Constants('ModelConstants.xls','Portfolio Constants','b12')
	~	dmnl
	~		|

Total IC=
	Total Novice IC+Total Intermediate IC+Total Expert IC
	~	people
	~		|

Total Intermediate IC=
	Total Intermediates*Base Intermediate Effectiveness
	~	people
	~		|

Total Expert IC=
	Total Experts*Base Expert Effectiveness
	~	people
	~		|

Total Novice IC=
	Total Novices*Base Novice Effectiveness
	~	people
	~		|

Total Control Rate=
	NR Control+IR Control+ER Control+NH Control+IH Control+EH Control+ND Control+ID Control\
		+ED Control+NT Control+IT Control+ET Control
	~	people
	~		|

Total Control PersonMonths= INTEG (
	Total Control Rate,
		0)
	~	people*months
	~		|

Total P1 Rate=
	P1NR+P1IR+P1ER+P1NH+P1IH+P1EH+P1ND+P1ID+P1ED+P1NT+P1IT+P1ET
	~	people
	~		|

Total P2 PersonMonths= INTEG (
	Total P2 Rate,
		0)
	~	people*months
	~		|

Total P2 Rate=
	P2NR+P2IR+P2ER+P2NH+P2IH+P2EH+P2ND+P2ID+P2ED+P2NT+P2IT+P2ET
	~	people
	~		|

Total P3 PersonMonths= INTEG (
	Total P3 Rate,
		0)
	~	people*months
	~		|
```

Total P3 Rate=
P3NR+P3IR+P3ER+P3NH+P3IH+P3EH+P3ND+P3ID+P3ED+P3NT+P3IT+P3ET
~ people
~ |

Total P4 personMonths= INTEG (
Total P4 Rate,
0)
~ people*months
~ |

Total P1 PersonMonths= INTEG (
Total P1 Rate,
0)
~ people*months
~ |

Total P4 Rate=
P4NR+P4IR+P4ER+P4NH+P4IH+P4EH+P4ND+P4ID+P4ED+P4NT+P4IT+P4ET
~ people
~ |

Total Novices=
TotalNR + TotalNH + TotalND + TotalNT
~ people
~ |

Total Intermediates=
TotalIR + TotalIH + TotalID + TotalIT
~ people
~ |

Total Experts=
TotalER + TotalEH + TotalED + TotalET
~ people
~ |

Total People=
Total Novices+Total Intermediates+Total Experts
~ people
~ |

Complexity Effect on Attrition T f(
[(0,0)-(10,10)],(0,1.2),(5,1),(10,0.7))
~ fraction
~ |

Complexity Effect on Attrition H f(
[(0,0)-(10,10)],(0,1.2),(5,1),(10,0.7))
~ fraction
~ |

Complexity Effect on Attrition D f(
[(0,0)-(10,10)],(0,1.2),(5,1),(10,0.7))
~ fraction
~ |

Complexity Effect on Attrition R f(
[(0,0)-(10,10)],(0,1.2),(5,1),(10,0.7))
~ fraction
~ |

Total PersonMonths R=
TotalNR Months + TotalIR Months + TotalER Months
~ people*months
~ |

Total PersonMonths T=
TotalNT Months + TotalIT Months + TotalET Months
~ people*months
~ |

Total PersonMonths D=

```
	TotalND Months + TotalID Months + TotalED Months
	~	people*months
	~		|

Total PersonMonths=
	Total PersonMonths D+Total PersonMonths H+Total PersonMonths R+Total PersonMonths T
	~	people*months
	~		|

Total PersonMonths H=
	TotalNH Months + TotalIH Months + TotalEH Months
	~	people*months
	~		|

TotalED Rate=
	TotalED
	~	people
	~		|

TotalNR Months= INTEG (
	TotalNR Rate,
		0)
	~	people*months
	~		|

TotalER Months= INTEG (
	TotalER Rate,
		0)
	~	people*months
	~		|

TotalNR Rate=
	TotalNR
	~	people
	~		|

TotalID Months= INTEG (
	TotalID Rate,
		0)
	~	people*months
	~		|

TotalNT Months= INTEG (
	TotalNT Rate,
		0)
	~	people*months
	~		|

TotalNT Rate=
	TotalNT
	~	people
	~		|

TotalIH Months= INTEG (
	TotalIH Rate,
		0)
	~	people*months
	~		|

TotalIH Rate=
	TotalIH
	~	people
	~		|

TotalED Months= INTEG (
	TotalED Rate,
		0)
	~	people*months
	~		|

TotalIR Rate=
	TotalIR
	~	people
```

```
	~		|

TotalND Rate=
	TotalND
	~	people
	~		|

TotalEH Months= INTEG (
	TotalEH Rate,
		0)
	~	people*months
	~		|

TotalEH Rate=
	TotalEH
	~	people
	~		|

TotalNH Rate=
	TotalNH
	~	people
	~		|

TotalER Rate=
	TotalER
	~	people
	~		|

TotalET Rate=
	TotalET
	~	people
	~		|

TotalET Months= INTEG (
	TotalET Rate,
		0)
	~	people*months
	~		|

TotalID Rate=
	TotalID
	~	people
	~		|

TotalIR Months= INTEG (
	TotalIR Rate,
		0)
	~	people*months
	~		|

PDY P1H=
	Complexity effect on PDY P1*Fatigue effect PDY P1H*Normal Productivity H
	~	lines/(people*Month)
	~		|

TotalIT Rate=
	TotalIT
	~	people
	~		|

TotalNH Months= INTEG (
	TotalNH Rate,
		0)
	~	people*months
	~		|

TotalND Months= INTEG (
	TotalND Rate,
		0)
	~	people*months
	~		|

TotalIT Months= INTEG (
```

TotalIT Rate,
0)
~ people*months
~ |

IntMultiplier P1H=
Base Intermediate Effectiveness*IN Ratio effect on effectiveness f(IN Ratio P1H)
~ fraction
~ |

IntMultiplier P1T=
Base Intermediate Effectiveness*IN Ratio effect on effectiveness f(IN Ratio P1T)
~ fraction
~ |

IntMultiplier P2D=
Base Intermediate Effectiveness*IN Ratio effect on effectiveness f(IN Ratio P2D)
~ fraction
~ |

IntMultiplier P2H=
Base Intermediate Effectiveness*IN Ratio effect on effectiveness f(IN Ratio P2H)
~ fraction
~ |

IntMultiplier P2R=
Base Intermediate Effectiveness*IN Ratio effect on effectiveness f(IN Ratio P2R)
~ fraction
~ |

IntMultiplier P2T=
Base Intermediate Effectiveness*IN Ratio effect on effectiveness f(IN Ratio P2T)
~ fraction
~ |

IntMultiplier P3D=
Base Intermediate Effectiveness*IN Ratio effect on effectiveness f(IN Ratio P3D)
~ fraction
~ |

IntMultiplier P3H=
Base Intermediate Effectiveness*IN Ratio effect on effectiveness f(IN Ratio P3H)
~ fraction
~ |

IntMultiplier P3R=
Base Intermediate Effectiveness*IN Ratio effect on effectiveness f(IN Ratio P3R)
~ fraction
~ |

IntMultiplier P3T=
Base Intermediate Effectiveness*IN Ratio effect on effectiveness f(IN Ratio P3T)
~ fraction
~ |

IntMultiplier P4D=
Base Intermediate Effectiveness*IN Ratio effect on effectiveness f(IN Ratio P4D)
~ fraction
~ |

IntMultiplier P4H=
Base Intermediate Effectiveness*IN Ratio effect on effectiveness f(IN Ratio P4H)
~ fraction
~ |

IntMultiplier P4R=
Base Intermediate Effectiveness*IN Ratio effect on effectiveness f(IN Ratio P4R)
~ fraction
~ |

IntMultiplier P4T=
Base Intermediate Effectiveness*IN Ratio effect on effectiveness f(IN Ratio P4T)
~ fraction

~ |

ExpertMultiplier P4D=
Base Expert Effectiveness*EI Ratio effect on effectiveness f(EI Ratio P4D)
~ fraction
~ |

NoviceMultiplier P1D=
Base Novice Effectiveness
~ fraction
~ |

NoviceMultiplier P1H=
Base Novice Effectiveness
~ fraction
~ |

NoviceMultiplier P3T=
Base Novice Effectiveness
~ fraction
~ |

NoviceMultiplier P1T=
Base Novice Effectiveness
~ fraction
~ |

NoviceMultiplier P2D=
Base Novice Effectiveness
~ fraction
~ |

NoviceMultiplier P2H=
Base Novice Effectiveness
~ fraction
~ |

NoviceMultiplier P2R=
Base Novice Effectiveness
~ fraction
~ |

NoviceMultiplier P2T=
Base Novice Effectiveness
~ fraction
~ |

NoviceMultiplier P3D=
Base Novice Effectiveness
~ fraction
~ |

NoviceMultiplier P3H=
Base Novice Effectiveness
~ fraction
~ |

NoviceMultiplier P3R=
Base Novice Effectiveness
~ fraction
~ |

ExpertMultiplier P1T=
Base Expert Effectiveness*EI Ratio effect on effectiveness f(EI Ratio P1T)
~ fraction
~ |

NoviceMultiplier P4D=
Base Novice Effectiveness
~ fraction
~ |

NoviceMultiplier P4H=

Base Novice Effectiveness
~ fraction
~ |

NoviceMultiplier P4R=
Base Novice Effectiveness
~ fraction
~ |

NoviceMultiplier P4T=
Base Novice Effectiveness
~ fraction
~ |

ExpertMultiplier P4R=
Base Expert Effectiveness*EI Ratio effect on effectiveness f(EI Ratio P4R)
~ fraction
~ |

ExpertMultiplier P4H=
Base Expert Effectiveness*EI Ratio effect on effectiveness f(EI Ratio P4H)
~ fraction
~ |

ExpertMultiplier P3T=
Base Expert Effectiveness*EI Ratio effect on effectiveness f(EI Ratio P3T)
~ fraction
~ |

ExpertMultiplier P4T=
Base Expert Effectiveness*EI Ratio effect on effectiveness f(EI Ratio P4T)
~ fraction
~ |

ExpertMultiplier P2T=
Base Expert Effectiveness*EI Ratio effect on effectiveness f(EI Ratio P2T)
~ fraction
~ |

ExpertMultiplier P2R=
Base Expert Effectiveness*EI Ratio effect on effectiveness f(EI Ratio P2R)
~ fraction
~ |

ExpertMultiplier P2H=
Base Expert Effectiveness*EI Ratio effect on effectiveness f(EI Ratio P2H)
~ fraction
~ |

IntMultiplier P1D=
Base Intermediate Effectiveness*IN Ratio effect on effectiveness f(IN Ratio P1D)
~ fraction
~ |

ExpertMultiplier P1D=
Base Expert Effectiveness*EI Ratio effect on effectiveness f(EI Ratio P1D)
~ fraction
~ |

ExpertMultiplier P3R=
Base Expert Effectiveness*EI Ratio effect on effectiveness f(EI Ratio P3R)
~ fraction
~ |

ExpertMultiplier P3H=
Base Expert Effectiveness*EI Ratio effect on effectiveness f(EI Ratio P3H)
~ fraction
~ |

ExpertMultiplier P1H=
Base Expert Effectiveness*EI Ratio effect on effectiveness f(EI Ratio P1H)
~ fraction
~ |

ExpertMultiplier P2D=
Base Expert Effectiveness*EI Ratio effect on effectiveness f(EI Ratio P2D)
~ fraction
~ |

ExpertMultiplier P3D=
Base Expert Effectiveness*EI Ratio effect on effectiveness f(EI Ratio P3D)
~ fraction
~ |

IN Ratio P1H=
ZIDZ(P1IH,P1NH)
~ fraction
~ |

IN Ratio P1R=
ZIDZ(P1IR,P1NR)
~ fraction
~ |

IN Ratio P1T=
ZIDZ(P1IT,P1NT)
~ fraction
~ |

IN Ratio P2D=
ZIDZ(P2ID,P2ND)
~ fraction
~ |

IN Ratio P2H=
ZIDZ(P2IH,P2NH)
~ fraction
~ |

IN Ratio P2R=
ZIDZ(P2IR,P2NR)
~ fraction
~ |

IN Ratio P2T=
ZIDZ(P2IT,P2NT)
~ fraction
~ |

IN Ratio P3D=
ZIDZ(P3ID,P3ND)
~ fraction
~ |

IN Ratio P3H=
ZIDZ(P3IH,P3NH)
~ fraction
~ |

IN Ratio P3R=
ZIDZ(P3IR,P3NR)
~ fraction
~ |

IN Ratio P3T=
ZIDZ(P3IT,P3NT)
~ fraction
~ |

IN Ratio P4D=
ZIDZ(P4ID,P4ND)
~ fraction
~ |

IN Ratio P4H=
ZIDZ(P4IH,P4NH)

```
	~	fraction
	~		|

IN Ratio P4R=
	ZIDZ(P4IR,P4NR)
	~	fraction
	~		|

IN Ratio P4T=
	ZIDZ(P4IT,P4NT)
	~	fraction
	~		|

EI Ratio P4R=
	ZIDZ(P4ER,P4IR)
	~	fraction
	~		|

EI Ratio P4H=
	ZIDZ(P4EH,P4IH)
	~	fraction
	~		|

EI Ratio P4D=
	ZIDZ(P4ED,P4ID)
	~	fraction
	~		|

EI Ratio P4T=
	ZIDZ(P4ET,P4IT)
	~	fraction
	~		|

EI Ratio P1T=
	ZIDZ(P1ET,P1IT)
	~	fraction
	~		|

EI Ratio P2T=
	ZIDZ(P2ET,P2IT)
	~	fraction
	~		|

EI Ratio P3R=
	ZIDZ(P3ER,P3IR)
	~	fraction
	~		|

EI Ratio effect on effectiveness f(
	[(0,0)-(1e+010,10)],(0,0),(0.01,0.1),(0.05,0.5),(0.1,0.9),(0.25,1),(0.5,1),(1,1),(10\
		,1),(100,1),(1e+010,1))
	~	fraction
	~		|

EI Ratio P1D=
	ZIDZ(P1ED,P1ID)
	~	fraction
	~		|

EI Ratio P1H=
	ZIDZ(P1EH,P1IH)
	~	fraction
	~		|

EI Ratio P1R=
	ZIDZ(P1ER,P1IR)
	~	fraction
	~		|

EI Ratio P2R=
	ZIDZ(P2ER,P2IR)
	~	fraction
	~		|
```

EI Ratio P2H=
ZIDZ(P2EH,P2IH)
~ fraction
~ |

EI Ratio P2D=
ZIDZ(P2ED,P2ID)
~ fraction
~ |

IN Ratio P1D=
ZIDZ(P1ID,P1ND)
~ fraction
~ |

Base Novice Effectiveness=
Get XLS Constants('ModelConstants.xls','Portfolio Constants','b17')
~ fraction
~ |

NoviceMultiplier P1R=
Base Novice Effectiveness
~ fraction
~ |

EI Ratio P3D=
ZIDZ(P3ED,P3ID)
~ fraction
~ |

EI Ratio P3T=
ZIDZ(P3ET,P3IT)
~ fraction
~ |

ExpertMultiplier P1R=
Base Expert Effectiveness*EI Ratio effect on effectiveness f(EI Ratio P1R)
~ fraction
~ |

IN Ratio effect on effectiveness f(
[(0,0)-(1000,10)],(0,0),(0.01,0.1),(0.05,0.5),(0.1,0.9),(0.25,1),(0.5,1),(1,1),(10,1\
),(1000,1))
~ fraction
~ |

Base Expert Effectiveness=
Get XLS Constants('ModelConstants.xls','Portfolio Constants','b19')
~ fraction
~ |

IntMultiplier P1R=
Base Intermediate Effectiveness*IN Ratio effect on effectiveness f(IN Ratio P1R)
~ fraction
~ |

EI Ratio P3H=
ZIDZ(P3EH,P3IH)
~ fraction
~ |

Base Intermediate Effectiveness=
Get XLS Constants('ModelConstants.xls','Portfolio Constants','b18')
~ fraction
~ |

TotalP2 H=
P2NH+P2IH+P2EH
~ people
~ |

AverageWorkerWeight P2H=

(RatioNH*NoviceMultiplier P2H+RatioIH*IntMultiplier P2H+RatioEH*ExpertMultiplier P2H\
)
~ fraction
~ |

AverageWorkerWeight P2R=
(RatioNR*NoviceMultiplier P2R+RatioIR*IntMultiplier P2R+RatioER*ExpertMultiplier P2R\
)
~ fraction
~ |

AverageWorkerWeight P2T=
(RatioNT*NoviceMultiplier P2T+RatioIT*IntMultiplier P2T+RatioET*ExpertMultiplier P2T\
)
~ fraction
~ |

AverageWorkerWeight P3D=
(RatioND*NoviceMultiplier P3D+RatioID*IntMultiplier P3D+RatioED*ExpertMultiplier P3D\
)
~ fraction
~ |

AverageWorkerWeight P3H=
(RatioNH*NoviceMultiplier P3H+RatioIH*IntMultiplier P3H+RatioEH*ExpertMultiplier P3H\
)
~ fraction
~ |

AverageWorkerWeight P3R=
(RatioNR*NoviceMultiplier P3R+RatioIR*IntMultiplier P3R+RatioER*ExpertMultiplier P3R\
)
~ fraction
~ |

AverageWorkerWeight P3T=
(RatioNT*NoviceMultiplier P3T+RatioIT*IntMultiplier P3T+RatioET*ExpertMultiplier P3T\
)
~ fraction
~ |

AverageWorkerWeight P4D=
(RatioND*NoviceMultiplier P4D+RatioID*IntMultiplier P4D+RatioED*ExpertMultiplier P4D\
)
~ fraction
~ |

AverageWorkerWeight P4H=
(RatioNH*NoviceMultiplier P4H+RatioIH*IntMultiplier P4H+RatioEH*ExpertMultiplier P4H\
)
~ fraction
~ |

AverageWorkerWeight P4R=
(RatioNR*NoviceMultiplier P4R+RatioIR*IntMultiplier P4R+RatioER*ExpertMultiplier P4R\
)
~ fraction
~ |

AverageWorkerWeight P4T=
(RatioNT*NoviceMultiplier P4T+RatioIT*IntMultiplier P4T+RatioET*ExpertMultiplier P4T\
)
~ fraction
~ |

DesiredRealHeads P1D=
DesiredPeople P1D/AverageWorkerWeight P1D
~ people
~ |

DesiredRealHeads P1H=
DesiredPeople P1H/AverageWorkerWeight P1H
~ people

```
	~			|

RatioED=
	ZIDZ(TotalED,(TotalND + TotalID + TotalED))
	~		fraction
	~			|

RatioEH=
	ZIDZ(TotalEH,(TotalNH + TotalIH + TotalEH))
	~		fraction
	~			|

RatioER=
	ZIDZ(TotalER,(TotalNR + TotalIR + TotalER))
	~		fraction
	~			|

RatioET=
	ZIDZ(TotalET,(TotalNT + TotalIT + TotalET))
	~		fraction
	~			|

RatioID=
	ZIDZ(TotalID,(TotalND + TotalID + TotalED))
	~		fraction
	~			|

RatioIH=
	ZIDZ(TotalIH,(TotalNH + TotalIH + TotalEH))
	~		fraction
	~			|

RatioIR=
	ZIDZ(TotalIR,(TotalNR + TotalIR + TotalER))
	~		fraction
	~			|

RatioIT=
	ZIDZ(TotalIT,(TotalIT + TotalIT + TotalET))
	~		fraction
	~			|

RatioND=
	ZIDZ(TotalND,(TotalND + TotalID + TotalED))
	~		fraction
	~			|

RatioNH=
	ZIDZ(TotalNH,(TotalNH + TotalIH + TotalEH))
	~		fraction
	~			|

RatioNR=
	ZIDZ(TotalNR,(TotalNR + TotalIR + TotalER))
	~		fraction
	~			|

RatioNT=
	ZIDZ(TotalNT,(TotalNT + TotalIT + TotalET))
	~		fraction
	~			|

DesiredRealHeads P4R=
	DesiredPeople P4R/AverageWorkerWeight P4R
	~		people
	~			|

DesiredRealHeads P4T=
	DesiredPeople P4T/AverageWorkerWeight P4T
	~		people
	~			|

AverageWorkerWeight P2D=
```

(RatioND*NoviceMultiplier P2D+RatioID*IntMultiplier P2D+RatioED*ExpertMultiplier P2D\
)
~ fraction
~ |

DesiredRealHeads P2T=
DesiredPeople P2T/AverageWorkerWeight P2T
~ people
~ |

DesiredRealHeads P3D=
DesiredPeople P3D/AverageWorkerWeight P3D
~ people
~ |

DesiredRealHeads P3H=
DesiredPeople P3H/AverageWorkerWeight P3H
~ people
~ |

P1RDesiredE=
DesiredRealHeads P1R*RatioER
~ people
~ |

P1RDesiredI=
DesiredRealHeads P1R*RatioIR
~ people
~ |

P1RDesiredN=
DesiredRealHeads P1R*RatioNR
~ people
~ |

AverageWorkerWeight P1R=
(RatioNR*NoviceMultiplier P1R+RatioIR*IntMultiplier P1R+RatioER*ExpertMultiplier P1R\
)
~ fraction
~ |

AverageWorkerWeight P1T=
(RatioNT*NoviceMultiplier P1T+RatioIT*IntMultiplier P1T+RatioET*ExpertMultiplier P1T\
)
~ fraction
~ |

DesiredRealHeads P2R=
DesiredPeople P2R/AverageWorkerWeight P2R
~ people
~ |

DesiredRealHeads P3T=
DesiredPeople P3T/AverageWorkerWeight P3T
~ people
~ |

DesiredRealHeads P4D=
DesiredPeople P4D/AverageWorkerWeight P4D
~ people
~ |

DesiredRealHeads P4H=
DesiredPeople P4H/AverageWorkerWeight P4H
~ people
~ |

DesiredRealHeads P3R=
DesiredPeople P3R/AverageWorkerWeight P3R
~ people
~ |

DesiredPeople P1H=

```
	((WorkToDo P1H/Remaining Time P1H)/Percvd PDY P1H+((Initial WorkToDo P1H/Remaining Time P1H\
		)/Percvd PDY P1H)*0.75*Active P1H)
	~	people
	~		|

DesiredRealHeads P1R=
	DesiredPeople P1R/AverageWorkerWeight P1R
	~	people
	~		|

DesiredRealHeads P1T=
	DesiredPeople P1T/AverageWorkerWeight P1T
	~	people
	~		|

DesiredRealHeads P2H=
	DesiredPeople P2H/AverageWorkerWeight P2H
	~	people
	~		|

AverageWorkerWeight P1H=
	(RatioNH*NoviceMultiplier P1H+RatioIH*IntMultiplier P1H+RatioEH*ExpertMultiplier P1H\
		)
	~	fraction
	~		|

DesiredRealHeads P2D=
	DesiredPeople P2D/AverageWorkerWeight P2D
	~	people
	~		|

AverageWorkerWeight P1D=
	(RatioND*NoviceMultiplier P1D+RatioID*IntMultiplier P1D+RatioED*ExpertMultiplier P1D\
		)
	~	fraction
	~		|

Total H=
	TotalNH + TotalIH + TotalEH
	~	people
	~		|

Total R=
	TotalNR + TotalIR + TotalER
	~	people
	~		|

Total T=
	TotalNT + TotalIT + TotalET
	~	people
	~		|

Total Desired R=
	SUM(NRDesired[project!])+SUM(IRDesired[project!])+SUM(ERDesired[project!])
	~	people
	~		|

Total D=
	TotalND + TotalID + TotalED
	~	people
	~		|

Total Desired D=
	SUM(NDDesired[project!])+SUM(IDDesired[project!])+SUM(EDDesired[project!])
	~	people
	~		|

Total Desired H=
	SUM(NHDesired[project!])+SUM(IHDesired[project!])+SUM(EHDesired[project!])
	~	people
	~		|

Total Desired T=
```

```
	SUM(NTDesired[project!])+SUM(ITDesired[project!])+SUM(ETDesired[project!])
	~	people
	~		|

Maximum Staff D=
	Get XLS Constants('ModelConstants.xls','Phase Constants','d10')
	~	people
	~		|

Maximum Staff H=
	Get XLS Constants('ModelConstants.xls','Phase Constants','c10')
	~	people
	~		|

Maximum Staff R=
	Get XLS Constants('ModelConstants.xls','Phase Constants','b10')
	~	people
	~		|

Maximum Staff T=
	Get XLS Constants('ModelConstants.xls','Phase Constants','e10')
	~	people
	~		|

NHHireRate=
	if then else(Novices to Hire H > 0, Novices to Hire H/(Time to hire/Gap Effect on Hiring H f\
		(GapRatio H)),0)
	~	people/Month
	~		|

Novices to Hire H=
	Min((Maximum Staff H-TotalNH-TotalIH-TotalEH),(SUM(NHDesired[project!]) +SUM(IHDesired\
		[project!])+SUM(EHDesired[project!]) - TotalNH-TotalIH-TotalNH))
	~	people
	~		|

NRHireRate=
	if then else(Novices to Hire R > 0, Novices to Hire R/(Time to hire/Gap Effect on Hiring R f\
		(GapRatio R)),0)
	~	people/Month
	~		|

NTHireRate=
	if then else(Novices to Hire T > 0, Novices to Hire T/(Time to hire/Gap Effect on Hiring T f\
		(GapRatio T)),0)
	~	people/Month
	~		|

NDHireRate=
	if then else(Novices to Hire D > 0, Novices to Hire D/(Time to hire/Gap Effect on Hiring D f\
		(GapRatio D)),0)
	~	people/Month
	~		|

Novices to Hire D=
	Min((Maximum Staff D-TotalND-TotalID-TotalED),(SUM(NDDesired[project!]) +SUM(IDDesired\
		[project!])+SUM(EDDesired[project!]) - TotalND-TotalID-TotalND))
	~	people
	~		|

Time to hire=
	Get XLS Constants('ModelConstants.xls','Portfolio Constants','b11')
	~	Month
	~		|

Novices to Hire T=
	Min((Maximum Staff T-TotalNT-TotalIT-TotalET),(SUM(NTDesired[project!]) +SUM(ITDesired\
		[project!])+SUM(ETDesired[project!]) - TotalNT-TotalIT-TotalNT))
	~	people
	~		|

Novices to Hire R=
	Min((Maximum Staff R-TotalNR-TotalIR-TotalER),(SUM(NRDesired[project!]) +SUM(IRDesired\
```

```
            [project!])+SUM(ERDesired[project!]) - TotalNR-TotalIR-TotalNR))
        ~       people
        ~               |

Complexity effect on PDY P1=
        Complexity effect on PDY f(Complexity P1)
        ~       fraction
        ~               |

Complexity effect on PDY P2=
        Complexity effect on PDY f(Complexity P2)
        ~       fraction
        ~               |

Complexity effect on PDY P3=
        Complexity effect on PDY f(Complexity P3)
        ~       fraction
        ~               |

Complexity effect on PDY P4=
        Complexity effect on PDY f(Complexity P4)
        ~       fraction
        ~               |

Complexity effect on quality f(
        [(0,0)-(10,10)],(0,1),(5,1),(10,0.8))
        ~       fraction
        ~               |

Complexity effect on quality P1=
        Complexity effect on quality f(Complexity P1)
        ~       fraction
        ~               |

Complexity effect on quality P2=
        Complexity effect on quality f(Complexity P2)
        ~       fraction
        ~               |

Complexity effect on quality P3=
        Complexity effect on quality f(Complexity P3)
        ~       fraction
        ~               |

Complexity effect on quality P4=
        Complexity effect on quality f(Complexity P4)
        ~       fraction
        ~               |

Qual P1R=
        Min(1, MaxQuality R*Fatigue effect qual P1R*Average Skill Effect on Quality P1R*Complexity effect on quality P1\
            )
        ~       fraction
        ~               |

Complexity P2=
        Get XLS Constants('ModelConstants.xls','Project Constants','x3')
        ~       dmnl
        ~               |

Complexity P3=
        Get XLS Constants('ModelConstants.xls','Project Constants','x4')
        ~       dmnl
        ~               |

Complexity P4=
        Get XLS Constants('ModelConstants.xls','Project Constants','x5')
        ~       dmnl
        ~               |

Complexity Weight=
        Get XLS Constants('ModelConstants.xls','Portfolio Constants','b5')
        ~       dmnl
```

~ |

Complexity effect on learning P2=
Complexity effect on learning f(Complexity P2)
~ fraction
~ |

Complexity effect on learning P3=
Complexity effect on learning f(Complexity P3)
~ fraction
~ |

Complexity effect on learning P4=
Complexity effect on learning f(Complexity P4)
~ fraction
~ |

Complexity effect on PDY f(
[(0,0)-(10,10)],(0,1.15),(1,1.13),(10,0.9605))
~ fraction
~ |

Complexity effect on attractiveness P2=
Complexity effect on attractiveness f(Complexity P2)
~ fraction
~ |

Complexity effect on attractiveness P3=
Complexity effect on attractiveness f(Complexity P3)
~ fraction
~ |

PDY P1R=
Normal Productivity R*Fatigue effect PDY P1R*Complexity effect on PDY P1
~ lines/(people*Month)
~ |

Complexity effect on learning f(
[(0,0)-(10,10)],(0,0.5),(5,1),(6,1.1),(10,1.5))
~ fraction
~ |

Complexity effect on learning P1=
Complexity effect on learning f(Complexity P1)
~ fraction
~ |

Attractiveness P1R=
(Bug ratio effect on attractiveness P1R*Bug Ratio Weight+Priority effect on attractiveness P1R\
*Priority Weight +Staffing Gap effect on attractiveness P1R
*Staffing Gap Weight + Complexity effect on attractiveness P1*Complexity Weight)*Active P1R
~ dmnl
~ |

Complexity effect on attractiveness f(
[(0,0)-(10,10)],(0,0.1),(5,0.5),(10,1))
~ fraction
~ |

Complexity effect on attractiveness P1=
Complexity effect on attractiveness f(Complexity P1)
~ fraction
~ |

Complexity P1=
Get XLS Constants('ModelConstants.xls','Project Constants','x2')
~ dmnl
~ |

Complexity effect on attractiveness P4=
Complexity effect on attractiveness f(Complexity P4)
~ fraction
~ |

```
Downsize Rate IH=
	if then else(Downsize Goal IH > 0, Min(Downsize Goal IH, IH Control)/Time to downsize\
		,0)
	~	people/Month
	~		|

Downsize Rate IR=
	if then else(Downsize Goal IR > 0, Min(Downsize Goal IR, IR Control)/Time to downsize\
		,0)
	~	people/Month
	~		|

Downsize Rate IT=
	if then else(Downsize Goal IT > 0, Min(Downsize Goal IT, IT Control)/Time to downsize\
		,0)
	~	people/Month
	~		|

NH Control= INTEG (
	-P1NH Rate - P2NH Rate - P3NH Rate - P4NH Rate+NHHireRate-Downsize Rate NH,
		0)
	~	people
	~		|

Qual P2D=
	Min(1, MaxQuality D*Fatigue effect qual P2D*Average Skill Effect on Quality P2D*Complexity effect on quality P2\
		)
	~	fraction
	~		|

Qual P2H=
	Min(1, MaxQuality H*Fatigue effect qual P2H*Average Skill Effect on Quality P2H*Complexity effect on quality P2\
		)
	~	fraction
	~		|

Qual P2R=
	Min(1, MaxQuality R*Fatigue effect qual P2R*Average Skill Effect on Quality P2R*Complexity effect on quality P2\
		)
	~	fraction
	~		|

Fatigue P2D=
	SMOOTHI(OverTime P2D,TimeToGetFatigued D,1)
	~	fraction
	~		|

Fatigue P2H=
	SMOOTHI(OverTime P2H,TimeToGetFatigued H,1)
	~	fraction
	~		|

Fatigue P2R=
	SMOOTHI(OverTime P2R,TimeToGetFatigued R,1)
	~	fraction
	~		|

Fatigue P2T=
	SMOOTHI(OverTime P2T,TimeToGetFatigued T,1)
	~	fraction
	~		|

Fatigue P3D=
	SMOOTHI(OverTime P3D,TimeToGetFatigued D,1)
	~	fraction
	~		|

Fatigue P3H=
	SMOOTHI(OverTime P3H,TimeToGetFatigued H,1)
	~	fraction
	~		|
```

```
Fatigue P3R=
	SMOOTHI(OverTime P3R,TimeToGetFatigued R,1)
	~	fraction
	~		|

Fatigue P3T=
	SMOOTHI(OverTime P3T,TimeToGetFatigued T,1)
	~	fraction
	~		|

Fatigue P4D=
	SMOOTHI(OverTime P4D,TimeToGetFatigued D,1)
	~	fraction
	~		|

Downsize Goal ED=
	TotalED - SUM(EDDesired[project!])
	~	people
	~		|

Downsize Goal EH=
	TotalEH - SUM(EHDesired[project!])
	~	people
	~		|

Downsize Goal ER=
	TotalER - SUM(ERDesired[project!])
	~	people
	~		|

Downsize Goal ET=
	TotalET - SUM(ETDesired[project!])
	~	people
	~		|

Downsize Goal ID=
	TotalID - SUM(IDDesired[project!])
	~	people
	~		|

Downsize Goal IH=
	TotalIH - SUM(IHDesired[project!])
	~	people
	~		|

Downsize Goal IR=
	TotalIR - SUM(IRDesired[project!])
	~	people
	~		|

Downsize Goal IT=
	TotalIT - SUM(ITDesired[project!])
	~	people
	~		|

Downsize Goal ND=
	TotalND - SUM(NDDesired[project!])
	~	people
	~		|

Downsize Goal NH=
	TotalNH - SUM(NHDesired[project!])
	~	people
	~		|

Downsize Goal NR=
	TotalNR - SUM(NRDesired[project!])
	~	people
	~		|

Downsize Goal NT=
	TotalNT - SUM(NTDesired[project!])
	~	people
```

```
	~		|

Downsize Rate ED=
	if then else(Downsize Goal ED > 0, Min(Downsize Goal ED, ED Control)/Time to downsize\
		,0)
	~	people/Month
	~		|

Downsize Rate EH=
	if then else(Downsize Goal EH > 0, Min(Downsize Goal EH, EH Control)/Time to downsize\
		,0)
	~	people/Month
	~		|

Downsize Rate ER=
	if then else(Downsize Goal ER > 0, Min(Downsize Goal ER, ER Control)/Time to downsize\
		,0)
	~	people/Month
	~		|

Downsize Rate ET=
	if then else(Downsize Goal ET > 0, Min(Downsize Goal ET, ET Control)/Time to downsize\
		,0)
	~	people/Month
	~		|

Downsize Rate ID=
	if then else(Downsize Goal ID > 0, Min(Downsize Goal ID, ID Control)/Time to downsize\
		,0)
	~	people/Month
	~		|

FindBugs P4H=
	HiddenBugs P4H/BugFindTime H
	~	lines/Month
	~		|

FindBugs P4R=
	HiddenBugs P4R/BugFindTime R
	~	lines/Month
	~		|

FindBugs P4T=
	HiddenBugs P4T/BugFindTime T
	~	lines/Month
	~		|

IT Control= INTEG (
	-P1IT Rate - P2IT Rate - P3IT Rate - P4IT Rate-Downsize Rate IT,
		0)
	~	people
	~		|

Percvd PDY P2R= INTEG (
	(PDY P2R - Percvd PDY P2R)/TimeToPercvPDY R,
		Normal Productivity R)
	~	lines/(people*Month)
	~		|

Percvd PDY P2T= INTEG (
	(PDY P2T - Percvd PDY P2T)/TimeToPercvPDY T,
		Normal Productivity T)
	~	lines/(people*Month)
	~		|

Percvd PDY P3D= INTEG (
	(PDY P3D - Percvd PDY P3D)/TimeToPercvPDY D,
		Normal Productivity D)
	~	lines/(people*Month)
	~		|

Intermediate Advance to Expert Time D=
	Get XLS Constants('ModelConstants.xls','Phase Constants','d3')
```

```
	~	Month
	~		|

Intermediate Advance to Expert Time H=
	Get XLS Constants('ModelConstants.xls','Phase Constants','c3')
	~	Month
	~		|

DueDate P2H=
	if then else(Time>(InitialDueDate P2H-minimum remaining time H),Time+minimum remaining time H\
		,InitialDueDate P2H)
	~	Month
	~		|

Intermediate Advance to Expert Time T=
	Get XLS Constants('ModelConstants.xls','Phase Constants','e3')
	~	Month
	~		|

DueDate P2T=
	if then else(Time>(InitialDueDate P2T-minimum remaining time T),Time+minimum remaining time T\
		,InitialDueDate P2T)
	~	Month
	~		|

DueDate P3D=
	if then else(Time>(InitialDueDate P3D-minimum remaining time D),Time+minimum remaining time D\
		,InitialDueDate P3D)
	~	Month
	~		|

DueDate P3H=
	if then else(Time>(InitialDueDate P3H-minimum remaining time H),Time+minimum remaining time H\
		,InitialDueDate P3H)
	~	Month
	~		|

IR Control= INTEG (
	-P1IR Rate - P2IR Rate - P3IR Rate - P4IR Rate-Downsize Rate IR,
		0)
	~	people
	~		|

DueDate P3T=
	if then else(Time>(InitialDueDate P3T-minimum remaining time T),Time+minimum remaining time T\
		,InitialDueDate P3T)
	~	Month
	~		|

DueDate P4D=
	if then else(Time>(InitialDueDate P4D-minimum remaining time D),Time+minimum remaining time D\
		,InitialDueDate P4D)
	~	Month
	~		|

DueDate P4H=
	if then else(Time>(InitialDueDate P4H-minimum remaining time H),Time+minimum remaining time H\
		,InitialDueDate P4H)
	~	Month
	~		|

Staffing Gap effect on learning P1T=
	Staffing Gap effect on learning f(ZIDZ(Workforce P1T,DesiredPeople P1T))
	~	dmnl
	~		|

DueDate P4T=
	if then else(Time>(InitialDueDate P4T-minimum remaining time T),Time+minimum remaining time T\
		,InitialDueDate P4T)
	~	Month
	~		|

ED Control= INTEG (
```

```
	-P1ED Rate - P2ED Rate - P3ED Rate - P4ED Rate-EDRetireRate-Downsize Rate ED,
		0)
	~	people
	~		|

Staffing Gap effect on learning P2T=
	Staffing Gap effect on learning f(ZIDZ(Workforce P2T,DesiredPeople P2T))
	~	dmnl
	~		|

Staffing Gap effect on learning P3D=
	Staffing Gap effect on learning f(ZIDZ(Workforce P3D,DesiredPeople P3D))
	~	dmnl
	~		|

Staffing Gap effect on learning P3H=
	Staffing Gap effect on learning f(ZIDZ(Workforce P3H,DesiredPeople P3H))
	~	dmnl
	~		|

Staffing Gap effect on learning P3R=
	Staffing Gap effect on learning f(ZIDZ(Workforce P3R,DesiredPeople P3R))
	~	dmnl
	~		|

Staffing Gap effect on learning P3T=
	Staffing Gap effect on learning f(ZIDZ(Workforce P3T,DesiredPeople P3T))
	~	dmnl
	~		|

Staffing Gap effect on learning P4D=
	Staffing Gap effect on learning f(ZIDZ(Workforce P4D,DesiredPeople P4D))
	~	dmnl
	~		|

Staffing Gap effect on learning P4H=
	Staffing Gap effect on learning f(ZIDZ(Workforce P4H,DesiredPeople P4H))
	~	dmnl
	~		|

Staffing Gap effect on learning P4R=
	Staffing Gap effect on learning f(ZIDZ(Workforce P4R,DesiredPeople P4R))
	~	dmnl
	~		|

Staffing Gap effect on learning P4T=
	Staffing Gap effect on learning f(ZIDZ(Workforce P4T,DesiredPeople P4T))
	~	dmnl
	~		|

ND Control= INTEG (
	-P1ND Rate - P2ND Rate - P3ND Rate - P4ND Rate+NDHireRate-Downsize Rate ND,
		0)
	~	people
	~		|

Qual P4T=
	Min(1, MaxQuality T*Fatigue effect qual P4T*Average Skill Effect on Quality P4T*Complexity effect on quality P4\
		)
	~	fraction
	~		|

Fatigue P4H=
	SMOOTHI(OverTime P4H,TimeToGetFatigued H,1)
	~	fraction
	~		|

ER Control= INTEG (
	-P1ER Rate - P2ER Rate - P3ER Rate - P4ER Rate-ERRetireRate-Downsize Rate ER,
		0)
	~	people
	~		|
```

```
Fatigue P4T=
	 SMOOTHI(OverTime P4T,TimeToGetFatigued T,1)
	~	fraction
	~		|

Qual P3R=
	Min(1, MaxQuality R*Fatigue effect qual P3R*Average Skill Effect on Quality P3R*Complexity effect on quality P3\
		)
	~	fraction
	~		|

Staffing Gap effect on learning P2D=
	Staffing Gap effect on learning f(ZIDZ(Workforce P2D,DesiredPeople P2D))
	~	dmnl
	~		|

Staffing Gap effect on learning P2H=
	Staffing Gap effect on learning f(ZIDZ(Workforce P2H,DesiredPeople P2H))
	~	dmnl
	~		|

Percvd PDY P2D= INTEG (
	(PDY P2D - Percvd PDY P2D)/TimeToPercvPDY D,
		Normal Productivity D)
	~	lines/(people*Month)
	~		|

Percvd PDY P2H= INTEG (
	(PDY P2H - Percvd PDY P2H)/TimeToPercvPDY H,
		Normal Productivity H)
	~	lines/(people*Month)
	~		|

Qual P2T=
	Min(1, MaxQuality T*Fatigue effect qual P2T*Average Skill Effect on Quality P2T*Complexity effect on quality P2\
		)
	~	fraction
	~		|

Qual P3D=
	Min(1, MaxQuality D*Fatigue effect qual P3D*Average Skill Effect on Quality P3D*Complexity effect on quality P3\
		)
	~	fraction
	~		|

Qual P3H=
	Min(1, MaxQuality H*Fatigue effect qual P3H*Average Skill Effect on Quality P3H*Complexity effect on quality P3\
		)
	~	fraction
	~		|

Staffing Gap effect on learning P1R=
	Staffing Gap effect on learning f(ZIDZ(Workforce P1R,DesiredPeople P1R))
	~	dmnl
	~		|

Novice Advance to Intermediate Time D=
	Get XLS Constants('ModelConstants.xls','Phase Constants','d2')
	~	Month
	~		|

Novice Advance to Intermediate Time H=
	Get XLS Constants('ModelConstants.xls','Phase Constants','c2')
	~	Month
	~		|

Qual P4H=
	Min(1, MaxQuality H*Fatigue effect qual P4H*Average Skill Effect on Quality P4H*Complexity effect on quality P4\
		)
	~	fraction
	~		|

Novice Advance to Intermediate Time T=
```

```
	Get XLS Constants('ModelConstants.xls','Phase Constants','e2')
	~	Month
	~		|

Staffing Gap effect on learning f(
	[(0,0)-(1e+009,2)],(0,0),(0.25,0.25),(0.5,0.5),(0.75,0.75),(1,1),(1.1,1.25),(2,1.5),\
		(1e+009,2))
	~	dmnl
	~		|

Staffing Gap effect on learning P1D=
	Staffing Gap effect on learning f(ZIDZ(Workforce P1D,DesiredPeople P1D))
	~	dmnl
	~		|

Staffing Gap effect on learning P1H=
	Staffing Gap effect on learning f(ZIDZ(Workforce P1H,DesiredPeople P1H))
	~	dmnl
	~		|

FindBugs P2T=
	 HiddenBugs P2T/BugFindTime T
	~	lines/Month
	~		|

NT Control= INTEG (
	-P1NT Rate - P2NT Rate - P3NT Rate - P4NT Rate+NTHireRate-Downsize Rate NT,
		0)
	~	people
	~		|

FindBugs P3H=
	 HiddenBugs P3H/BugFindTime H
	~	lines/Month
	~		|

DueDate P4R=
	if then else(Time>(InitialDueDate P4R-minimum remaining time R),Time+minimum remaining time R\
		,InitialDueDate P4R)
	~	Month
	~		|

NR Control= INTEG (
	-P1NR Rate - P2NR Rate - P3NR Rate - P4NR Rate+NRHireRate-Downsize Rate NR,
		0)
	~	people
	~		|

ET Control= INTEG (
	-P1ET Rate - P2ET Rate - P3ET Rate - P4ET Rate-ETRetireRate-Downsize Rate ET,
		0)
	~	people
	~		|

FindBugs P2H=
	 HiddenBugs P2H/BugFindTime H
	~	lines/Month
	~		|

FindBugs P2R=
	 HiddenBugs P2R/BugFindTime R
	~	lines/Month
	~		|

DueDate P3R=
	if then else(Time>(InitialDueDate P3R-minimum remaining time R),Time+minimum remaining time R\
		,InitialDueDate P3R)
	~	Month
	~		|

Fatigue P4R=
	 SMOOTHI(OverTime P4R,TimeToGetFatigued R,1)
	~	fraction
```

~ |

DueDate P2D=
if then else(Time>(InitialDueDate P2D-minimum remaining time D),Time+minimum remaining time D\
,InitialDueDate P2D)
~ Month
~ |

FindBugs P3R=
HiddenBugs P3R/BugFindTime R
~ lines/Month
~ |

FindBugs P3T=
HiddenBugs P3T/BugFindTime T
~ lines/Month
~ |

FindBugs P4D=
HiddenBugs P4D/BugFindTime D
~ lines/Month
~ |

Qual P4R=
Min(1, MaxQuality R*Fatigue effect qual P4R*Average Skill Effect on Quality P4R*Complexity effect on quality P4\
)
~ fraction
~ |

Percvd PDY P4R= INTEG (
(PDY P4R - Percvd PDY P4R)/TimeToPercvPDY R,
Normal Productivity R)
~ lines/(people*Month)
~ |

FindBugs P2D=
HiddenBugs P2D/BugFindTime D
~ lines/Month
~ |

Percvd PDY P3H= INTEG (
(PDY P3H - Percvd PDY P3H)/TimeToPercvPDY H,
Normal Productivity H)
~ lines/(people*Month)
~ |

Percvd PDY P4T= INTEG (
(PDY P4T - Percvd PDY P4T)/TimeToPercvPDY T,
Normal Productivity T)
~ lines/(people*Month)
~ |

IH Control= INTEG (
-P1IH Rate - P2IH Rate - P3IH Rate - P4IH Rate-Downsize Rate IH,
0)
~ people
~ |

DueDate P2R=
if then else(Time>(InitialDueDate P2R-minimum remaining time R),Time+minimum remaining time R\
,InitialDueDate P2R)
~ Month
~ |

Staffing Gap effect on learning P2R=
Staffing Gap effect on learning f(ZIDZ(Workforce P2R,DesiredPeople P2R))
~ dmnl
~ |

Percvd PDY P4H= INTEG (
(PDY P4H - Percvd PDY P4H)/TimeToPercvPDY H,
Normal Productivity H)
~ lines/(people*Month)

~ |

ID Control= INTEG (
-P1ID Rate - P2ID Rate - P3ID Rate - P4ID Rate-Downsize Rate ID,
0)
~ people
~ |

FindBugs P3D=
HiddenBugs P3D/BugFindTime D
~ lines/Month
~ |

Percvd PDY P4D= INTEG (
(PDY P4D - Percvd PDY P4D)/TimeToPercvPDY D,
Normal Productivity D)
~ lines/(people*Month)
~ |

Time to downsize=
Get XLS Constants('ModelConstants.xls','Portfolio Constants','b12')
~ Month
~ |

Qual P3T=
Min(1, MaxQuality T*Fatigue effect qual P3T*Average Skill Effect on Quality P3T*Complexity effect on quality P3\
)
~ fraction
~ |

Qual P4D=
Min(1, MaxQuality D*Fatigue effect qual P4D*Average Skill Effect on Quality P4D*Complexity effect on quality P4\
)
~ fraction
~ |

EH Control= INTEG (
-P1EH Rate - P2EH Rate - P3EH Rate - P4EH Rate-EHRetireRate-Downsize Rate EH,
0)
~ people
~ |

Percvd PDY P3R= INTEG (
(PDY P3R - Percvd PDY P3R)/TimeToPercvPDY R,
Normal Productivity R)
~ lines/(people*Month)
~ |

Percvd PDY P3T= INTEG (
(PDY P3T - Percvd PDY P3T)/TimeToPercvPDY T,
Normal Productivity T)
~ lines/(people*Month)
~ |

P3NT= INTEG (
P3NT Rate-P3NTtoIT Rate-Attrition Rate P3NT,
StartP3NT)
~ people
~ |

P2ET= INTEG (
P2ET Rate+P2ITtoET Rate-Attrition Rate P2ET,
StartP2ET)
~ people
~ |

Attrition Rate P4IT=
P4IT*Intermediate Attrition*Fatigue Effect on Attrition T f(Fatigue P4T)*Complexity Effect on Attrition T f\
(Complexity P4)/Time for Attrition
~ people/Month
~ |

Attrition Rate P3ET=

```
	P3ET*Expert Attrition*Fatigue Effect on Attrition T f(Fatigue P3T)*Complexity Effect on Attrition T f\
		(Complexity P3)/Time for Attrition
	~	people/Month
	~		|

P1IT= INTEG (
	P1IT Rate+P1NTtoIT Rate-P1ITtoET Rate-Attrition Rate P1IT,
		StartP1IT)
	~	people
	~		|

GapRatio T=
	ZIDZ((TotalNT+TotalIT+TotalET),(SUM(NTDesired[project!])+SUM(ITDesired[project!])+SUM\
		(ETDesired[project!])))
	~	fraction
	~		|

Gap Effect on Hiring T f(
	[(0,0)-(1e+009,20)],(0,2),(1e-005,2),(0.25,2),(0.5,1.5),(0.9,1),(1,0.9),(1.1,0.8),(2\
		,0.7),(10,0.2),(100,0.1),(1e+009,0.01))
	~	fraction
	~		|

P4NT= INTEG (
	P4NT Rate-P4NTtoIT Rate-Attrition Rate P4NT,
		StartP4NT)
	~	people
	~		|

Total Attrition ET= INTEG (
	Attrition Rate P1ET+Attrition Rate P2ET+Attrition Rate P3ET+Attrition Rate P4ET,
		0)
	~	people
	~		|

Fatigue Effect on Attrition T f(
	[(0,0)-(10,10)],(0,0.1),(0.776471,0.427046),(1.24706,0.782918),(1.64706,1.31673),(2,\
		1.9573))
	~	fraction
	~		|

P4ET= INTEG (
	P4ET Rate+P4ITtoET Rate-Attrition Rate P4ET,
		StartP4ET)
	~	people
	~		|

Attrition Rate P3NT=
	P3NT*Novice Attrition*Fatigue Effect on Attrition T f(Fatigue P3T)*Complexity Effect on Attrition T f\
		(Complexity P3)/Time for Attrition
	~	people/Month
	~		|

Total Attrition IT= INTEG (
	Attrition Rate P1IT+Attrition Rate P2IT+Attrition Rate P3IT+Attrition Rate P4IT,
		0)
	~	people
	~		|

Attrition Rate P1ET=
	P1ET*Expert Attrition*Fatigue Effect on Attrition T f(Fatigue P1T)*Complexity Effect on Attrition T f\
		(Complexity P1)/Time for Attrition
	~	people/Month
	~		|

Total Attrition NT= INTEG (
	Attrition Rate P1NT+Attrition Rate P2NT+Attrition Rate P3NT+Attrition Rate P4NT,
		0)
	~	people
	~		|

P1NT= INTEG (
	P1NT Rate-P1NTtoIT Rate-Attrition Rate P1NT,
```

```
		StartP1NT)
	~	people
	~		|

Attrition Rate P1IT=
	P1IT*Intermediate Attrition*Fatigue Effect on Attrition T f(Fatigue P1T)*Complexity Effect on Attrition T f\
		(Complexity P1)/Time for Attrition
	~	people/Month
	~		|

P1ET= INTEG (
	P1ET Rate+P1ITtoET Rate-Attrition Rate P1ET,
		StartP1ET)
	~	people
	~		|

P2NT= INTEG (
	P2NT Rate-P2NTtoIT Rate-Attrition Rate P2NT,
		StartP2NT)
	~	people
	~		|

Attrition Rate P4NT=
	P4NT*Novice Attrition*Fatigue Effect on Attrition T f(Fatigue P4T)*Complexity Effect on Attrition T f\
		(Complexity P4)/Time for Attrition
	~	people/Month
	~		|

Attrition Rate P1NT=
	P1NT*Novice Attrition*Fatigue Effect on Attrition T f(Fatigue P1T)*Complexity Effect on Attrition T f\
		(Complexity P1)/Time for Attrition
	~	people/Month
	~		|

TotalGap T=
	SUM(NTDesired[project!])-TotalNT+SUM(ITDesired[project!])-TotalIT+SUM(ETDesired[project\
		!])-TotalET
	~	people
	~		|

Attrition Rate P2ET=
	P2ET*Expert Attrition*Fatigue Effect on Attrition T f(Fatigue P2T)*Complexity Effect on Attrition T f\
		(Complexity P2)/Time for Attrition
	~	people/Month
	~		|

Attrition Rate P2IT=
	P2IT*Intermediate Attrition*Fatigue Effect on Attrition T f(Fatigue P2T)*Complexity Effect on Attrition T f\
		(Complexity P2)/Time for Attrition
	~	people/Month
	~		|

P3ET= INTEG (
	P3ET Rate+P3ITtoET Rate-Attrition Rate P3ET,
		StartP3ET)
	~	people
	~		|

Attrition Rate P2NT=
	P2NT*Novice Attrition*Fatigue Effect on Attrition T f(Fatigue P2T)*Complexity Effect on Attrition T f\
		(Complexity P2)/Time for Attrition
	~	people/Month
	~		|

P3IT= INTEG (
	P3IT Rate+P3NTtoIT Rate-P3ITtoET Rate-Attrition Rate P3IT,
		StartP3IT)
	~	people
	~		|

Attrition Rate P3IT=
	P3IT*Intermediate Attrition*Fatigue Effect on Attrition T f(Fatigue P3T)*Complexity Effect on Attrition T f\
		(Complexity P3)/Time for Attrition
```

```
	~	people/Month
	~		|

P2IT= INTEG (
	P2IT Rate+P2NTtoIT Rate-P2ITtoET Rate-Attrition Rate P2IT,
		StartP2IT)
	~	people
	~		|

Attrition Rate P4ET=
	P4ET*Expert Attrition*Fatigue Effect on Attrition T f(Fatigue P4T)*Complexity Effect on Attrition T f\
		(Complexity P4)/Time for Attrition
	~	people/Month
	~		|

P4IT= INTEG (
	P4IT Rate+P4NTtoIT Rate-P4ITtoET Rate-Attrition Rate P4IT,
		StartP4IT)
	~	people
	~		|

Active P1D=
	if then else((Initial WorkToDo P1D<Init Dev P1D):AND:(((Initial WorkToDo P1D+WorkToDo P1D\
		)/Init Dev P1D)>0.02),1,0)
	~	dmnl
	~		|

Active P1H=
	if then else((Initial WorkToDo P1H<Init HLD P1H):AND:(((Initial WorkToDo P1H+WorkToDo P1H\
		)/Init HLD P1H)>0.02),1,0)
	~	dmnl
	~		|

Active P1R=
	if then else((Time>TimeToStart P1R):AND:((WorkToDo P1R/InitialWorkToDo P1R)>0.02),1,\
		0)
	~	dmnl
	~		|

Active P1T=
	if then else((Initial WorkToDo P1T<Init Test P1T):AND:(((WorkToDo P1T+Initial WorkToDo P1T\
		)/Init Test P1T)>0.02),1,0)
	~	dmnl
	~		|

Active P2D=
	if then else((Initial WorkToDo P2D<Init Dev P2D):AND:(((Initial WorkToDo P2D+WorkToDo P2D\
		)/Init Dev P2D)>0.02),1,0)
	~	dmnl
	~		|

Active P2H=
	if then else((Initial WorkToDo P2H<Init HLD P2H):AND:(((Initial WorkToDo P2H+WorkToDo P2H\
		)/Init HLD P2H)>0.02),1,0)
	~	dmnl
	~		|

Active P2R=
	if then else((Time>TimeToStart P2R):AND:((WorkToDo P2R/InitialWorkToDo P2R)>0.02),1,\
		0)
	~	dmnl
	~		|

Active P2T=
	if then else((Initial WorkToDo P2T<Init Test P2T):AND:(((WorkToDo P2T+Initial WorkToDo P2T\
		)/Init Test P2T)>0.02),1,0)
	~	dmnl
	~		|

Active P3D=
	if then else((Initial WorkToDo P3D<Init Dev P3D):AND:(((Initial WorkToDo P3D+WorkToDo P3D\
		)/Init Dev P3D)>0.02),1,0)
	~	dmnl
```

```
	~		|

Active P3H=
	if then else((Initial WorkToDo P3H<Init HLD P3H):AND:(((Initial WorkToDo P3H+WorkToDo P3H\
		)/Init HLD P3H)>0.02),1,0)
	~	dmnl
	~		|

Active P3R=
	if then else((Time>TimeToStart P3R):AND:((WorkToDo P3R/InitialWorkToDo P3R)>0.02),1,\
		0)
	~	dmnl
	~		|

Active P3T=
	if then else((Initial WorkToDo P3T<Init Test P3T):AND:(((Initial WorkToDo P3T+WorkToDo P3T\
		)/Init Test P3T)>0.02),1,0)
	~	dmnl
	~		|

Active P4D=
	if then else((Initial WorkToDo P4D<Init Dev P4D):AND:(((Initial WorkToDo P4D+WorkToDo P4D\
		)/Init Dev P4D)>0.02),1,0)
	~	dmnl
	~		|

Active P4H=
	if then else((Initial WorkToDo P4H<Init HLD P4H):AND:(((Initial WorkToDo P4H+WorkToDo P4H\
		)/Init HLD P4H)>0.02),1,0)
	~	dmnl
	~		|

Active P4R=
	if then else((Time>TimeToStart P4R):AND:((WorkToDo P4R/InitialWorkToDo P4R)>0.02),1,\
		0)
	~	dmnl
	~		|

Active P4T=
	if then else((Initial WorkToDo P4T<Init Test P4T):AND:(((Initial WorkToDo P4T+WorkToDo P4T\
		)/Init Test P4T)>0.02),1,0)
	~	dmnl
	~		|

AllocatedP1ED=
	EDAllocated[one]
	~	people
	~		|

AllocatedP1EH=
	EHAllocated[one]
	~	people
	~		|

AllocatedP1ER=
	ERAllocated[one]
	~	people
	~		|

AllocatedP1ET=
	ETAllocated[one]
	~	people
	~		|

AllocatedP1ID=
	IDAllocated[one]
	~	people
	~		|

AllocatedP1IH=
	IHAllocated[one]
	~	people
	~		|
```

```
AllocatedP1IR=
	IRAllocated[one]
	~	people
	~		|

AllocatedP1IT=
	ITAllocated[one]
	~	people
	~		|

AllocatedP1ND=
	NDAllocated[one]
	~	people
	~		|

AllocatedP1NH=
	NHAllocated[one]
	~	people
	~		|

AllocatedP1NR=
	NRAllocated[one]
	~	people
	~		|

AllocatedP1NT=
	NTAllocated[one]
	~	people
	~		|

AllocatedP2ED=
	EDAllocated[two]
	~	people
	~		|

AllocatedP2EH=
	EHAllocated[two]
	~	people
	~		|

AllocatedP2ER=
	ERAllocated[two]
	~	people
	~		|

AllocatedP2ET=
	ETAllocated[two]
	~	people
	~		|

AllocatedP2ID=
	IDAllocated[two]
	~	people
	~		|

AllocatedP2IH=
	IHAllocated[two]
	~	people
	~		|

AllocatedP2IR=
	IRAllocated[two]
	~	people
	~		|

AllocatedP2IT=
	ITAllocated[two]
	~	people
	~		|

AllocatedP2ND=
	NDAllocated[two]
```

```
	~	people
	~		|

AllocatedP2NH=
	NHAllocated[two]
	~	people
	~		|

AllocatedP2NR=
	NRAllocated[two]
	~	people
	~		|

AllocatedP2NT=
	NTAllocated[two]
	~	people
	~		|

AllocatedP3ED=
	EDAllocated[three]
	~	people
	~		|

AllocatedP3EH=
	EHAllocated[three]
	~	people
	~		|

AllocatedP3ER=
	ERAllocated[three]
	~	people
	~		|

AllocatedP3ET=
	ETAllocated[three]
	~	people
	~		|

AllocatedP3ID=
	IDAllocated[three]
	~	people
	~		|

AllocatedP3IH=
	IHAllocated[three]
	~	people
	~		|

AllocatedP3IR=
	IRAllocated[three]
	~	people
	~		|

AllocatedP3IT=
	ITAllocated[three]
	~	people
	~		|

AllocatedP3ND=
	NDAllocated[three]
	~	people
	~		|

AllocatedP3NH=
	NHAllocated[three]
	~	people
	~		|

AllocatedP3NR=
	NRAllocated[three]
	~	people
	~		|
```

AllocatedP3NT=
NTAllocated[three]
~ people
~ |

AllocatedP4ED=
EDAllocated[four]
~ people
~ |

AllocatedP4EH=
EHAllocated[four]
~ people
~ |

AllocatedP4ER=
ERAllocated[four]
~ people
~ |

AllocatedP4ET=
ETAllocated[four]
~ people
~ |

AllocatedP4ID=
IDAllocated[four]
~ people
~ |

AllocatedP4IR=
IRAllocated[four]
~ people
~ |

AllocatedP4IR 0=
IHAllocated[four]
~ people
~ |

AllocatedP4IT=
ITAllocated[four]
~ people
~ |

AllocatedP4ND=
NDAllocated[four]
~ people
~ |

AllocatedP4NH=
NHAllocated[four]
~ people
~ |

AllocatedP4NT=
NTAllocated[four]
~ people
~ |

Attractiveness P1D=
(Bug ratio effect on attractiveness P1D*Bug Ratio Weight+Priority effect on attractiveness P1D\
*Priority Weight +Staffing Gap effect on attractiveness P1D
*Staffing Gap Weight+Complexity effect on attractiveness P1*Complexity Weight)*Active P1D
~ dmnl
~ |

Attractiveness P1H=
(Bug ratio effect on attractiveness P1H*Bug Ratio Weight+Priority effect on attractiveness P1H\
*Priority Weight +Staffing Gap effect on attractiveness P1H
*Staffing Gap Weight+Complexity effect on attractiveness P1*Complexity Weight)*Active P1H
~ dmnl
~ |

Attractiveness P1T=
(Bug ratio effect on attractiveness P1T*Bug Ratio Weight+Priority effect on attractiveness P1T\
*Priority Weight +Staffing Gap effect on attractiveness P1T
*Staffing Gap Weight+Complexity effect on attractiveness P1*Complexity Weight)*Active P1T
~ dmnl
~ |

Attractiveness P2D=
(Bug ratio effect on attractiveness P2D*Bug Ratio Weight+Priority effect on attractiveness P2D\
*Priority Weight +Staffing Gap effect on attractiveness P2D
*Staffing Gap Weight+ Complexity effect on attractiveness P2*Complexity Weight)*Active P2D
~ dmnl
~ |

Attractiveness P2H=
(Bug ratio effect on attractiveness P2H*Bug Ratio Weight+Priority effect on attractiveness P2H\
*Priority Weight +Staffing Gap effect on attractiveness P2H
*Staffing Gap Weight+ Complexity effect on attractiveness P2*Complexity Weight)*Active P2H
~ dmnl
~ |

Attractiveness P2R=
(Bug ratio effect on attractiveness P2R*Bug Ratio Weight+Priority effect on attractiveness P2R\
*Priority Weight +Staffing Gap effect on attractiveness P2R
*Staffing Gap Weight+ Complexity effect on attractiveness P2*Complexity Weight)*Active P2R
~ dmnl
~ |

Attractiveness P2T=
(Bug ratio effect on attractiveness P2T*Bug Ratio Weight+Priority effect on attractiveness P2T\
*Priority Weight +Staffing Gap effect on attractiveness P2T
*Staffing Gap Weight+ Complexity effect on attractiveness P2*Complexity Weight)*Active P2T
~ dmnl
~ |

Attractiveness P3D=
(Bug ratio effect on attractiveness P3D*Bug Ratio Weight+Priority effect on attractiveness P3D\
*Priority Weight +Staffing Gap effect on attractiveness P3D
*Staffing Gap Weight + Complexity effect on attractiveness P3*Complexity Weight)*Active P3D
~ dmnl
~ |

Attractiveness P3H=
(Bug ratio effect on attractiveness P3H*Bug Ratio Weight+Priority effect on attractiveness P3H\
*Priority Weight +Staffing Gap effect on attractiveness P3H
*Staffing Gap Weight + Complexity effect on attractiveness P3*Complexity Weight)*Active P3H
~ dmnl
~ |

Attractiveness P3R=
(Bug ratio effect on attractiveness P3R*Bug Ratio Weight+Priority effect on attractiveness P3R\
*Priority Weight +Staffing Gap effect on attractiveness P3R
*Staffing Gap Weight + Complexity effect on attractiveness P3*Complexity Weight)*Active P3R
~ dmnl
~ |

Attractiveness P4D=
(Bug ratio effect on attractiveness P4D*Bug Ratio Weight+Priority effect on attractiveness P4D\
*Priority Weight +Staffing Gap effect on attractiveness P4D
*Staffing Gap Weight + Complexity effect on attractiveness P4*Complexity Weight)*Active P4D
~ dmnl
~ |

Attractiveness P4H=
(Bug ratio effect on attractiveness P4H*Bug Ratio Weight+Priority effect on attractiveness P4H\
*Priority Weight +Staffing Gap effect on attractiveness P4H
*Staffing Gap Weight + Complexity effect on attractiveness P4*Complexity Weight)*Active P4H
~ dmnl
~ |

Attractiveness P4R=
(Bug ratio effect on attractiveness P4R*Bug Ratio Weight+Priority effect on attractiveness P4R\

*Priority Weight +Staffing Gap effect on attractiveness P4R
*Staffing Gap Weight+ Complexity effect on attractiveness P4*Complexity Weight)*Active P4R
~ dmnl
~ |

Attractiveness P4T=
(Bug ratio effect on attractiveness P4T*Bug Ratio Weight+Priority effect on attractiveness P4T\
*Priority Weight +Staffing Gap effect on attractiveness P4T
*Staffing Gap Weight + Complexity effect on attractiveness P4*Complexity Weight)*Active P4T
~ dmnl
~ |

Attrition Rate P1ED=
P1ED*Expert Attrition*Fatigue Effect on Attrition D f(Fatigue P1D)*Complexity Effect on Attrition D f\
(Complexity P1)/Time for Attrition
~ people/Month
~ |

Attrition Rate P1EH=
P1EH*Expert Attrition*Fatigue Effect on Attrition H f(Fatigue P1H)*Complexity Effect on Attrition H f\
(Complexity P1)/Time for Attrition
~ people/Month
~ |

Attrition Rate P1ER=
P1ER*Expert Attrition*Fatigue Effect on Attrition R f(Fatigue P1R)*Complexity Effect on Attrition R f\
(Complexity P1)/Time for Attrition
~ people/Month
~ |

Attrition Rate P1ID=
P1ID*Intermediate Attrition*Fatigue Effect on Attrition D f(Fatigue P1D)*Complexity Effect on Attrition D f\
(Complexity P1)/Time for Attrition
~ people/Month
~ |

Attrition Rate P1IH=
P1IH*Intermediate Attrition*Fatigue Effect on Attrition H f(Fatigue P1H)*Complexity Effect on Attrition H f\
(Complexity P1)/Time for Attrition
~ people/Month
~ |

Attrition Rate P1IR=
P1IR*Intermediate Attrition*Fatigue Effect on Attrition R f(Fatigue P1R)*Complexity Effect on Attrition R f\
(Complexity P1)/Time for Attrition
~ people/Month
~ |

Attrition Rate P1ND=
P1ND*Novice Attrition*Fatigue Effect on Attrition D f(Fatigue P1D)*Complexity Effect on Attrition D f\
(Complexity P1)/Time for Attrition
~ people/Month
~ |

Attrition Rate P1NH=
P1NH*Novice Attrition*Fatigue Effect on Attrition H f(Fatigue P1H)*Complexity Effect on Attrition H f\
(Complexity P1)/Time for Attrition
~ people/Month
~ |

Attrition Rate P1NR=
P1NR*Novice Attrition*Fatigue Effect on Attrition R f(Fatigue P1R)*Complexity Effect on Attrition R f\
(Complexity P1)/Time for Attrition
~ people/Month
~ |

Attrition Rate P2ED=
P2ED*Expert Attrition*Fatigue Effect on Attrition D f(Fatigue P2D)*Complexity Effect on Attrition D f\
(Complexity P2)/Time for Attrition
~ people/Month
~ |

Attrition Rate P2EH=

P2EH*Expert Attrition*Fatigue Effect on Attrition H f(Fatigue P2H)*Complexity Effect on Attrition H f\
(Complexity P2)/Time for Attrition
~ people/Month
~ |

Attrition Rate P2ER=
P2ER*Expert Attrition*Fatigue Effect on Attrition R f(Fatigue P2R)*Complexity Effect on Attrition R f\
(Complexity P2)/Time for Attrition
~ people/Month
~ |

Attrition Rate P2ID=
P2ID*Intermediate Attrition*Fatigue Effect on Attrition D f(Fatigue P2D)*Complexity Effect on Attrition D f\
(Complexity P2)/Time for Attrition
~ people/Month
~ |

Attrition Rate P2IH=
P2IH*Intermediate Attrition*Fatigue Effect on Attrition H f(Fatigue P2H)*Complexity Effect on Attrition H f\
(Complexity P2)/Time for Attrition
~ people/Month
~ |

Attrition Rate P2IR=
P2IR*Intermediate Attrition*Fatigue Effect on Attrition R f(Fatigue P2R)*Complexity Effect on Attrition R f\
(Complexity P2)/Time for Attrition
~ people/Month
~ |

Attrition Rate P2ND=
P2ND*Novice Attrition*Fatigue Effect on Attrition D f(Fatigue P2D)*Complexity Effect on Attrition D f\
(Complexity P2)/Time for Attrition
~ people/Month
~ |

Attrition Rate P2NH=
P2NH*Novice Attrition*Fatigue Effect on Attrition H f(Fatigue P2H)*Complexity Effect on Attrition H f\
(Complexity P2)/Time for Attrition
~ people/Month
~ |

Attrition Rate P2NR=
P2NR*Novice Attrition*Fatigue Effect on Attrition R f(Fatigue P2R)*Complexity Effect on Attrition R f\
(Complexity P2)/Time for Attrition
~ people/Month
~ |

Attrition Rate P3ED=
P3ED*Expert Attrition*Fatigue Effect on Attrition D f(Fatigue P3D)*Complexity Effect on Attrition D f\
(Complexity P3)/Time for Attrition
~ people/Month
~ |

Attrition Rate P3EH=
P3EH*Expert Attrition*Fatigue Effect on Attrition H f(Fatigue P3H)*Complexity Effect on Attrition H f\
(Complexity P3)/Time for Attrition
~ people/Month
~ |

Attrition Rate P3ER=
P3ER*Expert Attrition*Fatigue Effect on Attrition R f(Fatigue P3R)*Complexity Effect on Attrition R f\
(Complexity P3)/Time for Attrition
~ people/Month
~ |

Attrition Rate P3ID=
P3ID*Intermediate Attrition*Fatigue Effect on Attrition D f(Fatigue P3D)*Complexity Effect on Attrition D f\
(Complexity P3)/Time for Attrition
~ people/Month
~ |

Attrition Rate P3IH=
P3IH*Intermediate Attrition*Fatigue Effect on Attrition H f(Fatigue P3H)*Complexity Effect on Attrition H f\

(Complexity P3)/Time for Attrition
~ people/Month
~ |

Attrition Rate P3IR=
P3IR*Intermediate Attrition*Fatigue Effect on Attrition R f(Fatigue P3R)*Complexity Effect on Attrition R f\
(Complexity P3)/Time for Attrition
~ people/Month
~ |

Attrition Rate P3ND=
P3ND*Novice Attrition*Fatigue Effect on Attrition D f(Fatigue P3D)*Complexity Effect on Attrition D f\
(Complexity P3)/Time for Attrition
~ people/Month
~ |

Attrition Rate P3NH=
P3NH*Novice Attrition*Fatigue Effect on Attrition H f(Fatigue P3H)*Complexity Effect on Attrition H f\
(Complexity P3)/Time for Attrition
~ people/Month
~ |

Attrition Rate P3NR=
P3NR*Novice Attrition*Fatigue Effect on Attrition R f(Fatigue P3R)*Complexity Effect on Attrition R f\
(Complexity P3)/Time for Attrition
~ people/Month
~ |

Attrition Rate P4ED=
P4ED*Expert Attrition*Fatigue Effect on Attrition D f(Fatigue P4D)*Complexity Effect on Attrition D f\
(Complexity P4)/Time for Attrition
~ people/Month
~ |

Attrition Rate P4EH=
P4EH*Expert Attrition*Fatigue Effect on Attrition H f(Fatigue P4H)*Complexity Effect on Attrition H f\
(Complexity P4)/Time for Attrition
~ people/Month
~ |

Attrition Rate P4ER=
P4ER*Expert Attrition*Fatigue Effect on Attrition R f(Fatigue P4R)*Complexity Effect on Attrition R f\
(Complexity P4)/Time for Attrition
~ people/Month
~ |

Attrition Rate P4ID=
P4ID*Intermediate Attrition*Fatigue Effect on Attrition D f(Fatigue P4D)*Complexity Effect on Attrition D f\
(Complexity P4)/Time for Attrition
~ people/Month
~ |

Attrition Rate P4IH=
P4IH*Intermediate Attrition*Fatigue Effect on Attrition H f(Fatigue P4H)*Complexity Effect on Attrition H f\
(Complexity P4)/Time for Attrition
~ people/Month
~ |

Attrition Rate P4IR=
P4IR*Intermediate Attrition*Fatigue Effect on Attrition R f(Fatigue P4R)*Complexity Effect on Attrition R f\
(Complexity P4)/Time for Attrition
~ people/Month
~ |

Attrition Rate P4ND=
P4ND*Novice Attrition*Fatigue Effect on Attrition D f(Fatigue P4D)*Complexity Effect on Attrition D f\
(Complexity P4)/Time for Attrition
~ people/Month
~ |

Attrition Rate P4NH=
P4NH*Novice Attrition*Fatigue Effect on Attrition H f(Fatigue P4H)*Complexity Effect on Attrition H f\
(Complexity P4)/Time for Attrition

~ people/Month
~ |

Attrition Rate P4NR=
P4NR*Novice Attrition*Fatigue Effect on Attrition R f(Fatigue P4R)*Complexity Effect on Attrition R f\
(Complexity P4)/Time for Attrition
~ people/Month
~ |

Average Skill Effect on Quality P1D=
((P1ND*NoviceMultiplier P1D*Novice Skill Effect on Quality)+(P1ID*IntMultiplier P1D*\
Intermediate Skill Effect on Quality)+(P1ED*ExpertMultiplier P1D*Expert Skill Effect on Quality\
))/Workforce P1D
~ fraction
~ |

Average Skill Effect on Quality P4T=
((P4NT*NoviceMultiplier P4T*Novice Skill Effect on Quality)+(P4IT*IntMultiplier P4T*\
Intermediate Skill Effect on Quality)+(P4ET*ExpertMultiplier P4T*Expert Skill Effect on Quality\
))/Workforce P4T
~ fraction
~ |

Average Skill Effect on Quality P1H=
((P1NH*NoviceMultiplier P1H*Novice Skill Effect on Quality)+(P1IH*IntMultiplier P1H*\
Intermediate Skill Effect on Quality)+(P1EH*ExpertMultiplier P1H*Expert Skill Effect on Quality\
))/Workforce P1H
~ fraction
~ |

Average Skill Effect on Quality P1R=
((P1NR*NoviceMultiplier P1R*Novice Skill Effect on Quality)+(P1IR*IntMultiplier P1R*\
Intermediate Skill Effect on Quality)+(P1ER*ExpertMultiplier P1R*Expert Skill Effect on Quality\
))/Workforce P1R
~ fraction
~ |

Average Skill Effect on Quality P1T=
((P1NT*NoviceMultiplier P1T*Novice Skill Effect on Quality)+(P1IT*IntMultiplier P1T*\
Intermediate Skill Effect on Quality)+(P1ET*ExpertMultiplier P1T*Expert Skill Effect on Quality\
))/Workforce P1T
~ fraction
~ |

Average Skill Effect on Quality P2D=
((P2ND*NoviceMultiplier P2D*Novice Skill Effect on Quality)+(P2ID*IntMultiplier P2D*\
Intermediate Skill Effect on Quality)+(P2ED*ExpertMultiplier P2D*Expert Skill Effect on Quality\
))/Workforce P2D
~ fraction
~ |

Average Skill Effect on Quality P2H=
((P2NH*NoviceMultiplier P2H*Novice Skill Effect on Quality)+(P2IH*IntMultiplier P2H*\
Intermediate Skill Effect on Quality)+(P2EH*ExpertMultiplier P2H*Expert Skill Effect on Quality\
))/Workforce P2H
~ fraction
~ |

Average Skill Effect on Quality P2R=
((P2NR*NoviceMultiplier P2R*Novice Skill Effect on Quality)+(P2IR*IntMultiplier P2R*\
Intermediate Skill Effect on Quality)+(P2ER*ExpertMultiplier P2R*Expert Skill Effect on Quality\
))/Workforce P2R
~ fraction
~ |

Average Skill Effect on Quality P2T=
((P2NT*NoviceMultiplier P2T*Novice Skill Effect on Quality)+(P2IT*IntMultiplier P2T*\
Intermediate Skill Effect on Quality)+(P2ET*ExpertMultiplier P2T*Expert Skill Effect on Quality\
))/Workforce P2T
~ fraction
~ |

Average Skill Effect on Quality P3D=

```
	((P3ND*NoviceMultiplier P3D*Novice Skill Effect on Quality)+(P3ID*IntMultiplier P3D*\
		Intermediate Skill Effect on Quality)+(P3ED*ExpertMultiplier P3D*Expert Skill Effect on Quality\
		))/Workforce P3D
	~	fraction
	~		|

Average Skill Effect on Quality P3H=
	((P3NH*NoviceMultiplier P3H*Novice Skill Effect on Quality)+(P3IH*IntMultiplier P3H*\
		Intermediate Skill Effect on Quality)+(P3EH*ExpertMultiplier P3H*Expert Skill Effect on Quality\
		))/Workforce P3H
	~	fraction
	~		|

Average Skill Effect on Quality P3R=
	((P3NR*NoviceMultiplier P3R*Novice Skill Effect on Quality)+(P3IR*IntMultiplier P3R*\
		Intermediate Skill Effect on Quality)+(P3ER*ExpertMultiplier P3R*Expert Skill Effect on Quality\
		))/Workforce P3R
	~	fraction
	~		|

Average Skill Effect on Quality P3T=
	((P3NT*NoviceMultiplier P3T*Novice Skill Effect on Quality)+(P3IT*IntMultiplier P3T*\
		Intermediate Skill Effect on Quality)+(P3ET*ExpertMultiplier P3T*Expert Skill Effect on Quality\
		))/Workforce P3T
	~	fraction
	~		|

Average Skill Effect on Quality P4D=
	((P4ND*NoviceMultiplier P4D*Novice Skill Effect on Quality)+(P4ID*IntMultiplier P4D*\
		Intermediate Skill Effect on Quality)+(P4ED*ExpertMultiplier P4D*Expert Skill Effect on Quality\
		))/Workforce P4D
	~	fraction
	~		|

Average Skill Effect on Quality P4H=
	((P4NH*NoviceMultiplier P4H*Novice Skill Effect on Quality)+(P4IH*IntMultiplier P4H*\
		Intermediate Skill Effect on Quality)+(P4EH*ExpertMultiplier P4H*Expert Skill Effect on Quality\
		))/Workforce P4H
	~	fraction
	~		|

Average Skill Effect on Quality P4R=
	((P4NR*NoviceMultiplier P4R*Novice Skill Effect on Quality)+(P4IR*IntMultiplier P4R*\
		Intermediate Skill Effect on Quality)+(P4ER*ExpertMultiplier P4R*Expert Skill Effect on Quality\
		))/Workforce P4R
	~	fraction
	~		|

Bug ratio effect on attractiveness f(
	[(0,0)-(2e+030,1)],(0,1),(1,0),(100,0),(1e+030,0))
	~	dmnl
	~		|

Bug ratio effect on attractiveness P1D=
	Bug ratio effect on attractiveness f(ZIDZ(HiddenBugs P1D,Done Right P1D))
	~	dmnl
	~		|

Bug ratio effect on attractiveness P1H=
	Bug ratio effect on attractiveness f(ZIDZ(HiddenBugs P1H,Done Right P1H))
	~	dmnl
	~		|

Bug ratio effect on attractiveness P1R=
	Bug ratio effect on attractiveness f(ZIDZ(HiddenBugs P1R,Done Right P1R))
	~	dmnl
	~		|

Bug ratio effect on attractiveness P1T=
	Bug ratio effect on attractiveness f(ZIDZ(HiddenBugs P1T,Done Right P1T))
	~	dmnl
	~		|
```

Bug ratio effect on attractiveness P2D=
Bug ratio effect on attractiveness f(ZIDZ(HiddenBugs P2D,Done Right P2D))
~ dmnl
~ |

Bug ratio effect on attractiveness P2H=
Bug ratio effect on attractiveness f(ZIDZ(HiddenBugs P2H,Done Right P2H))
~ dmnl
~ |

Bug ratio effect on attractiveness P2R=
Bug ratio effect on attractiveness f(ZIDZ(HiddenBugs P2R,Done Right P2R))
~ dmnl
~ |

Bug ratio effect on attractiveness P2T=
Bug ratio effect on attractiveness f(ZIDZ(HiddenBugs P2T,Done Right P2T))
~ dmnl
~ |

Bug ratio effect on attractiveness P3D=
Bug ratio effect on attractiveness f(ZIDZ(HiddenBugs P3D,Done Right P3D))
~ dmnl
~ |

Bug ratio effect on attractiveness P3H=
Bug ratio effect on attractiveness f(ZIDZ(HiddenBugs P3H,Done Right P3H))
~ dmnl
~ |

Bug ratio effect on attractiveness P3R=
Bug ratio effect on attractiveness f(ZIDZ(HiddenBugs P3R,Done Right P3R))
~ dmnl
~ |

Bug ratio effect on attractiveness P3T=
Bug ratio effect on attractiveness f(ZIDZ(HiddenBugs P3T,Done Right P3T))
~ dmnl
~ |

Bug ratio effect on attractiveness P4D=
Bug ratio effect on attractiveness f(ZIDZ(HiddenBugs P4D,Done Right P4D))
~ dmnl
~ |

Bug ratio effect on attractiveness P4H=
Bug ratio effect on attractiveness f(ZIDZ(HiddenBugs P4H,Done Right P4H))
~ dmnl
~ |

Bug ratio effect on attractiveness P4R=
Bug ratio effect on attractiveness f(ZIDZ(HiddenBugs P4R,Done Right P4R))
~ dmnl
~ |

Bug ratio effect on attractiveness P4T=
Bug ratio effect on attractiveness f(ZIDZ(HiddenBugs P4T,Done Right P4T))
~ dmnl
~ |

Bug Ratio Weight=
Get XLS Constants('ModelConstants.xls','Portfolio Constants','b3')
~ dmnl
~ |

BugFindTime D=
Get XLS Constants('ModelConstants.xls','Phase Constants','d9')
~ Month
~ |

BugFindTime H=
Get XLS Constants('ModelConstants.xls','Phase Constants','c9')
~ Month

```
	~		|

BugFindTime R=
	Get XLS Constants('ModelConstants.xls','Phase Constants','b9')
	~	Month
	~		|

BugFindTime T=
	Get XLS Constants('ModelConstants.xls','Phase Constants','e9')
	~	Month
	~		|

DesiredP1ED=
	P1DDesiredE
	~	people
	~		|

DesiredP1EH=
	P1HDesiredE
	~	people
	~		|

DesiredP1ER=
	P1RDesiredE
	~	people
	~		|

DesiredP1ET=
	P1TDesiredE
	~	people
	~		|

DesiredP1ID=
	P1DDesiredI
	~	people
	~		|

DesiredP1IH=
	P1HDesiredI
	~	people
	~		|

DesiredP1IR=
	P1RDesiredI
	~	people
	~		|

DesiredP1IT=
	P1TDesiredI
	~	people
	~		|

DesiredP1ND=
	P1DDesiredN
	~	people
	~		|

DesiredP1NH=
	P1HDesiredN
	~	people
	~		|

DesiredP1NR=
	P1RDesiredN
	~	people
	~		|

DesiredP1NT=
	P1TDesiredN
	~	people
	~		|

DesiredP2ED=
```

P2DDesiredE
~ people
~ |

DesiredP2EH=
P2HDesiredE
~ people
~ |

DesiredP2ER=
P2RDesiredE
~ people
~ |

DesiredP2ET=
P2TDesiredE
~ people
~ |

DesiredP2ID=
P2DDesiredI
~ people
~ |

DesiredP2IH=
P2HDesiredI
~ people
~ |

DesiredP2IR=
P2RDesiredI .
~ people
~ |

DesiredP2IT=
P2TDesiredI
~ people
~ |

DesiredP2ND=
P2DDesiredN
~ people
~ |

DesiredP2NH=
P2HDesiredN
~ people
~ |

DesiredP2NR=
P2RDesiredN
~ people
~ |

DesiredP2NT=
P2TDesiredN
~ people
~ |

DesiredP3ED=
P3DDesiredE
~ people
~ |

DesiredP3EH=
P3HDesiredE
~ people
~ |

DesiredP3ER=
P3RDesiredE
~ people
~ |

DesiredP3ET=
P3TDesiredE
~ people
~ |

DesiredP3ID=
P3DDesiredI
~ people
~ |

DesiredP3IH=
P3HDesiredI
~ people
~ |

DesiredP3IR=
P3RDesiredI
~ people
~ |

DesiredP3IT=
P3TDesiredI
~ people
~ |

DesiredP3ND=
P3DDesiredN
~ people
~ |

DesiredP3NH=
P3HDesiredN
~ people
~ |

DesiredP3NR=
P3RDesiredN
~ people
~ |

DesiredP3NT=
P3TDesiredN
~ people
~ |

DesiredP4ED=
P4DDesiredE
~ people
~ |

DesiredP4EH=
P4HDesiredE
~ people
~ |

DesiredP4ER=
P4RDesiredE
~ people
~ |

DesiredP4ET=
P4TDesiredE
~ people
~ |

DesiredP4ID=
P4DDesiredI
~ people
~ |

DesiredP4IH=
P4HDesiredI

~ people
~ |

DesiredP4IR=
P4RDesiredI
~ people
~ |

DesiredP4IT=
P4TDesiredI
~ people
~ |

DesiredP4ND=
P4DDesiredN
~ people
~ |

DesiredP4NH=
P4HDesiredN
~ people
~ |

DesiredP4NR=
P4RDesiredN
~ people
~ |

DesiredP4NT=
P4TDesiredN
~ people
~ |

DesiredPeople P1D=
(WorkToDo P1D/Remaining Time P1D)/Percvd PDY P1D+((Initial WorkToDo P1D/Remaining Time P1D\
)/Percvd PDY P1D)*0.75*Active P1D
~ people
~ |

DesiredPeople P1R=
(WorkToDo P1R/Remaining Time P1R)/Percvd PDY P1R
~ people
~ |

DesiredPeople P1T=
(WorkToDo P1T/Remaining Time P1T)/Percvd PDY P1T+((Initial WorkToDo P1T/Remaining Time P1T\
)/Percvd PDY P1T)*0.75*Active P1T
~ people
~ |

DesiredPeople P2D=
(WorkToDo P2D/Remaining Time P2D)/Percvd PDY P2D+((Initial WorkToDo P2D/Remaining Time P2D\
)/Percvd PDY P2D)*0.75*Active P2D
~ people
~ |

DesiredPeople P2H=
((WorkToDo P2H/Remaining Time P2H)/Percvd PDY P2H+((Initial WorkToDo P2H/Remaining Time P2H\
)/Percvd PDY P2H)*0.75*Active P2H)
~ people
~ |

DesiredPeople P2R=
(WorkToDo P2R/Remaining Time P2R)/Percvd PDY P2R
~ people
~ |

DesiredPeople P2T=
(WorkToDo P2T/Remaining Time P2T)/Percvd PDY P2T+((Initial WorkToDo P2T/Remaining Time P2T\
)/Percvd PDY P2T)*0.75*Active P2T
~ people
~ |

DesiredPeople P3D=
(WorkToDo P3D/Remaining Time P3D)/Percvd PDY P3D+((Initial WorkToDo P3D/Remaining Time P3D\
)/Percvd PDY P3D)*0.75*Active P3D
~ people
~ |

DesiredPeople P3H=
if then else(P3 Initial Priority>0,((WorkToDo P3H/Remaining Time P3H)/Percvd PDY P3H\
+((Initial WorkToDo P3H/Remaining Time P3H)/Percvd PDY P3H)*0.75*Active P3H),0)
~ people
~ |

DesiredPeople P3R=
if then else(P3 Initial Priority>0, (WorkToDo P3R/Remaining Time P3R)/Percvd PDY P3R\
,0)
~ people
~ |

DesiredPeople P3T=
(WorkToDo P3T/Remaining Time P3T)/Percvd PDY P3T+((Initial WorkToDo P3T/Remaining Time P3T\
)/Percvd PDY P3T)*0.75*Active P3T
~ people
~ |

DesiredPeople P4D=
(WorkToDo P4D/Remaining Time P4D)/Percvd PDY P4D+((Initial WorkToDo P4D/Remaining Time P4D\
)/Percvd PDY P4D)*0.75*Active P4D
~ people
~ |

DesiredPeople P4H=
((WorkToDo P4H/Remaining Time P4H)/Percvd PDY P4H+((Initial WorkToDo P4H/Remaining Time P4H\
)/Percvd PDY P4H)*0.75*Active P4H)
~ people
~ |

DesiredPeople P4R=
(WorkToDo P4R/Remaining Time P4R)/Percvd PDY P4R
~ people
~ |

DesiredPeople P4T=
(WorkToDo P4T/Remaining Time P4T)/Percvd PDY P4T+((Initial WorkToDo P4T/Remaining Time P4T\
)/Percvd PDY P4T)*0.75*Active P4T
~ people
~ |

Doing P1D=
Min(WorkToDo P1D/TIME STEP,Effective People P1D*PDY P1D)
~ lines/Month
~ |

Doing P1H=
Min(WorkToDo P1H/TIME STEP,Effective People P1H*PDY P1H)
~ lines/Month
~ |

Doing P1R=
Min(WorkToDo P1R/TIME STEP,Effective People P1R*PDY P1R)
~ lines/Month
~ |

Doing P1T=
Min(WorkToDo P1T/TIME STEP,Effective People P1T*PDY P1T)
~ lines/Month
~ |

Doing P2D=
Min(WorkToDo P2D/TIME STEP,Effective People P2D*PDY P2D)
~ lines/Month
~ |

Doing P2H=

Min(WorkToDo P2H/TIME STEP,Effective People P2H*PDY P2H)
~ lines/Month
~ |

Doing P2R=
Min(WorkToDo P2R/TIME STEP,Effective People P2R*PDY P2R)
~ lines/Month
~ |

Doing P2T=
Min(WorkToDo P2T/TIME STEP,Effective People P2T*PDY P2T)
~ lines/Month
~ |

Doing P3D=
Min(WorkToDo P3D/TIME STEP,Effective People P3D*PDY P3D)
~ lines/Month
~ |

Doing P3H=
Min(WorkToDo P3H/TIME STEP,Effective People P3H*PDY P3H)
~ lines/Month
~ |

Doing P3R=
Min(WorkToDo P3R/TIME STEP,Effective People P3R*PDY P3R)
~ lines/Month
~ |

Doing P3T=
Min(WorkToDo P3T/TIME STEP,Effective People P3T*PDY P3T)
~ lines/Month
~ |

Doing P4D=
Min(WorkToDo P4D/TIME STEP,Effective People P4D*PDY P4D)
~ lines/Month
~ |

Doing P4H=
Min(WorkToDo P4H/TIME STEP,Effective People P4H*PDY P4H)
~ lines/Month
~ |

Doing P4R=
Min(WorkToDo P4R/TIME STEP,Effective People P4R*PDY P4R)
~ lines/Month
~ |

Doing P4T=
Min(WorkToDo P4T/TIME STEP,Effective People P4T*PDY P4T)
~ lines/Month
~ |

Doing right P1D=
Doing P1D*Qual P1D
~ lines/Month
~ |

Doing right P1H=
Doing P1H*Qual P1H
~ lines/Month
~ |

Doing right P1R=
Doing P1R*Qual P1R
~ lines/Month
~ |

Doing right P1T=
Doing P1T*Qual P1T
~ lines/Month
~ |

Doing right P2D=
Doing P2D*Qual P2D
~ lines/Month
~ |

Doing right P2H=
Doing P2H*Qual P2H
~ lines/Month
~ |

Doing right P2R=
Doing P2R*Qual P2R
~ lines/Month
~ |

Doing right P2T=
Doing P2T*Qual P2T
~ lines/Month
~ |

Doing right P3D=
Doing P3D*Qual P3D
~ lines/Month
~ |

Doing right P3H=
Doing P3H*Qual P3H
~ lines/Month
~ |

Doing right P3R=
Doing P3R*Qual P3R
~ lines/Month
~ |

Doing right P3T=
Doing P3T*Qual P3T
~ lines/Month
~ |

Doing right P4D=
Doing P4D*Qual P4D
~ lines/Month
~ |

Doing right P4H=
Doing P4H*Qual P4H
~ lines/Month
~ |

Doing right P4R=
Doing P4R*Qual P4R
~ lines/Month
~ |

Doing right P4T=
Doing P4T*Qual P4T
~ lines/Month
~ |

Doing wrong P1D = Doing P1D*(1-Qual P1D)
~ lines/Month
~ |

Doing wrong P1H = Doing P1H*(1-Qual P1H)
~ lines/Month
~ |

Doing wrong P1R = Doing P1R*(1-Qual P1R)
~ lines/Month
~ |

```
Doing wrong P1T = Doing P1T*(1-Qual P1T)
	~	lines/Month
	~		|

Doing wrong P2D = Doing P2D*(1-Qual P2D)
	~	lines/Month
	~		|

Doing wrong P2H = Doing P2H*(1-Qual P2H)
	~	lines/Month
	~		|

Doing wrong P2R = Doing P2R*(1-Qual P2R)
	~	lines/Month
	~		|

Doing wrong P2T = Doing P2T*(1-Qual P2T)
	~	lines/Month
	~		|

Doing wrong P3D = Doing P3D*(1-Qual P3D)
	~	lines/Month
	~		|

Doing wrong P3H = Doing P3H*(1-Qual P3H)
	~	lines/Month
	~		|

Doing wrong P3R = Doing P3R*(1-Qual P3R)
	~	lines/Month
	~		|

Doing wrong P3T = Doing P3T*(1-Qual P3T)
	~	lines/Month
	~		|

Doing wrong P4D = Doing P4D*(1-Qual P4D)
	~	lines/Month
	~		|

Doing wrong P4H = Doing P4H*(1-Qual P4H)
	~	lines/Month
	~		|

Doing wrong P4R = Doing P4R*(1-Qual P4R)
	~	lines/Month
	~		|

Doing wrong P4T = Doing P4T*(1-Qual P4T)
	~	lines/Month
	~		|

Done Right P1D= INTEG (
	Doing right P1D,
		0)
	~	lines
	~		|

Done Right P1H= INTEG (
	Doing right P1H,
		0)
	~	lines
	~		|

Done Right P1R= INTEG (
	Doing right P1R,
		0)
	~	lines
	~		|

Done Right P1T = INTEG(Doing right P1T,  0)
	~	lines
	~		|
```

```
Done Right P2D= INTEG (
	Doing right P2D,
		0)
	~	lines
	~		|

Done Right P2H= INTEG (
	Doing right P2H,
		0)
	~	lines
	~		|

Done Right P2R = INTEG(Doing right P2R,  0)
	~	lines
	~		|

Done Right P2T = INTEG(Doing right P2T,  0)
	~	lines
	~		|

Done Right P3D = INTEG(Doing right P3D,  0)
	~	lines
	~		|

Done Right P3H = INTEG(Doing right P3H,  0)
	~	lines
	~		|

Done Right P3R = INTEG(Doing right P3R,  0)
	~	lines
	~		|

Done Right P3T = INTEG(Doing right P3T,  0)
	~	lines
	~		|

Done Right P4D = INTEG(Doing right P4D,  0)
	~	lines
	~		|

Done Right P4H = INTEG(Doing right P4H,  0)
	~	lines
	~		|

Done Right P4R = INTEG(Doing right P4R,  0)
	~	lines
	~		|

Done Right P4T = INTEG(Doing right P4T,  0)
	~	lines
	~		|

DueDate P1D=
	if then else(Time>(InitialDueDate P1D-minimum remaining time D),Time+minimum remaining time D\
		,InitialDueDate P1D)
	~	Month
	~		|

DueDate P1H=
	if then else(Time>(InitialDueDate P1H-minimum remaining time H),Time+minimum remaining time H\
		,InitialDueDate P1H)
	~	Month
	~		|

DueDate P1R=
	if then else(Time>(InitialDueDate P1R-minimum remaining time R),Time+minimum remaining time R\
		,InitialDueDate P1R)
	~	Month
	~		|

DueDate P1T=
	if then else(Time>(InitialDueDate P1T-minimum remaining time T),Time+minimum remaining time T\
```

```
                    ,InitialDueDate P1T)
        ~           Month
        ~                   |

EDAllocated[project]=
        Allocate By Priority(EDDesired[project],EDAttractiveness[project],4,1,TotalED)
        ~           people
        ~                   |

EDAttractiveness[one]=
        Attractiveness P1D ~~|
EDAttractiveness[two]=
        Attractiveness P2D ~~|
EDAttractiveness[three]=
        Attractiveness P3D ~~|
EDAttractiveness[four]=
        Attractiveness P4D
        ~           dmnl
        ~                   |

EDDesired[one]=
        DesiredP1ED ~~|
EDDesired[two]=
        DesiredP2ED ~~|
EDDesired[three]=
        DesiredP3ED ~~|
EDDesired[four]=
        DesiredP4ED
        ~           people
        ~                   |

EDRetired= INTEG (
        EDRetireRate,
                    0)
        ~           people
        ~                   |

EDRetireRate=
        Min(Experts to retire D, ED Control) / Time to retire
        ~           people/Month
        ~                   |

Effective People P1D=
        OverTime P1D*Workforce P1D
        ~           people
        ~                   |

Effective People P1H=
        OverTime P1H*Workforce P1H
        ~           people
        ~                   |

Effective People P1R=
        OverTime P1R*Workforce P1R
        ~           people
        ~                   |

Effective People P1T=
        OverTime P1T*Workforce P1T
        ~           people
        ~                   |

Effective People P2D=
        OverTime P2D*Workforce P2D
        ~           people
        ~                   |

Effective People P2H=
        OverTime P2H*Workforce P2H
        ~           people
        ~                   |

Effective People P2R=
```

```
	OverTime P2R*Workforce P2R
	~	people
	~		|

Effective People P2T=
	OverTime P2T*Workforce P2T
	~	people
	~		|

Effective People P3D=
	OverTime P3D*Workforce P3D
	~	people
	~		|

Effective People P3H=
	OverTime P3H*Workforce P3H
	~	people
	~		|

Effective People P3R=
	OverTime P3R*Workforce P3R
	~	people
	~		|

Effective People P3T=
	OverTime P3T*Workforce P3T
	~	people
	~		|

Effective People P4D=
	OverTime P4D*Workforce P4D
	~	people
	~		|

Effective People P4H=
	OverTime P4H*Workforce P4H
	~	people
	~		|

Effective People P4R=
	OverTime P4R*Workforce P4R
	~	people
	~		|

Effective People P4T=
	OverTime P4T*Workforce P4T
	~	people
	~		|

EffectOfFatigue OnProductivity f(
	[(0,0)-(2,2)],(0,1.1),(0.5,1.1),(0.769231,1.06355),(1.00306,0.982456),(1.19266,0.877193\
		),(1.45566,0.736842),(1.98777,0.596491))
	~	dmnl
	~		|

EffectOfFatigue OnQuality f(
	[(0,0)-(2,1.1)],(0,1),(0.616314,1),(1,1),(1.30887,0.907018),(1.51682,0.839474),(1.72477\
		,0.767105),(1.98777,0.651316))
	~	dmnl
	~		|

EHAllocated[project]=
	Allocate By Priority(EHDesired[project],EHAttractiveness[project],4,1,TotalEH)
	~	people
	~		|

EHAttractiveness[one]=
	Attractiveness P1H ~~|
EHAttractiveness[two]=
	Attractiveness P2H ~~|
EHAttractiveness[three]=
	Attractiveness P3H ~~|
EHAttractiveness[four]=
```

```
	Attractiveness P4H
	~	dmnl
	~		|

EHDesired[one]=
	DesiredP1EH ~~|
EHDesired[two]=
	DesiredP2EH ~~|
EHDesired[three]=
	DesiredP3EH ~~|
EHDesired[four]=
	DesiredP4EH
	~	people
	~		|

EHRetired= INTEG (
	EHRetireRate,
		0)
	~	people
	~		|

EHRetireRate=
	Min(Experts to retire H, EH Control)/Time to retire
	~	people/Month
	~		|

ERAllocated[project]=
	Allocate By Priority(ERDesired[project],ERAttractiveness[project],4,1,TotalER)
	~	people
	~		|

ERAttractiveness[one]=
	Attractiveness P1R ~~|
ERAttractiveness[two]=
	Attractiveness P2R ~~|
ERAttractiveness[three]=
	Attractiveness P3R ~~|
ERAttractiveness[four]=
	Attractiveness P4R
	~	dmnl
	~		|

ERDesired[one]=
	DesiredP1ER ~~|
ERDesired[two]=
	DesiredP2ER ~~|
ERDesired[three]=
	DesiredP3ER ~~|
ERDesired[four]=
	DesiredP4ER
	~	people
	~		|

ERRetired= INTEG (
	ERRetireRate,
		0)
	~	people
	~		|

ERRetireRate=
	Min(Experts to retire R, ER Control)/Time to retire
	~	people/Month
	~		|

ETAllocated[project]=
	Allocate By Priority(ETDesired[project],ETAttractiveness[project],4,1,TotalET)
	~	people
	~		|

ETAttractiveness[one]=
	Attractiveness P1T ~~|
ETAttractiveness[two]=
	Attractiveness P2T ~~|
```

```
ETAttractiveness[three]=
	Attractiveness P3T ~~|
ETAttractiveness[four]=
	Attractiveness P4T
	~	dmnl
	~		|

ETDesired[one]=
	DesiredP1ET ~~|
ETDesired[two]=
	DesiredP2ET ~~|
ETDesired[three]=
	DesiredP3ET ~~|
ETDesired[four]=
	DesiredP4ET
	~	people
	~		|

ETRetired= INTEG (
	ETRetireRate,
		0)
	~	people
	~		|

ETRetireRate=
	Min(Experts to retire T, ET Control)/Time to retire
	~	people/Month
	~		|

Expert Attrition=
	0.1
	~	fraction
	~	10 percent / year
	|

Expert Skill Effect on Quality=
	Get XLS Constants('ModelConstants.xls','Portfolio Constants','b22')
	~	fraction
	~		|

Experts to retire D=
	0.02*TotalED
	~	people
	~		|

Experts to retire H=
	0.02*TotalEH
	~	people
	~		|

Experts to retire R=
	0.01*TotalER
	~	people
	~	0.02*TotalER
	|

Experts to retire T=
	0.02*TotalET
	~	people
	~		|

Fatigue Effect on Attrition D f(
	[(0,0)-(10,10)],(0,0.1),(0.776471,0.427046),(1.24706,0.782918),(1.64706,1.31673),(2,\
		1.9573))
	~	fraction
	~		|

Fatigue Effect on Attrition H f(
	[(0,0)-(10,10)],(0,0.1),(0.776471,0.427046),(1.24706,0.782918),(1.64706,1.31673),(2,\
		1.9573))
	~	fraction
	~		|
```

Fatigue Effect on Attrition R f(
[(0,0)-(10,10)],(0,0.1),(0.776471,0.427046),(1.24706,0.782918),(1.64706,1.31673),(2,\
1.9573))
~ fraction
~ |

Fatigue effect PDY P1D=
EffectOfFatigue OnProductivity f(Fatigue P1D)
~ dmnl
~ |

Fatigue effect PDY P1H=
EffectOfFatigue OnProductivity f(Fatigue P1H)
~ dmnl
~ |

Fatigue effect PDY P1R=
EffectOfFatigue OnProductivity f(Fatigue P1R)
~ dmnl
~ |

Fatigue effect PDY P1T=
EffectOfFatigue OnProductivity f(Fatigue P1T)
~ dmnl
~ |

Fatigue effect PDY P2D=
EffectOfFatigue OnProductivity f(Fatigue P2D)
~ dmnl
~ |

Fatigue effect PDY P2H=
EffectOfFatigue OnProductivity f(Fatigue P2H)
~ dmnl
~ |

Fatigue effect PDY P2R=
EffectOfFatigue OnProductivity f(Fatigue P2R)
~ dmnl
~ |

Fatigue effect PDY P2T=
EffectOfFatigue OnProductivity f(Fatigue P2T)
~ dmnl
~ |

Fatigue effect PDY P3D=
EffectOfFatigue OnProductivity f(Fatigue P3D)
~ dmnl
~ |

Fatigue effect PDY P3H=
EffectOfFatigue OnProductivity f(Fatigue P3H)
~ dmnl
~ |

Fatigue effect PDY P3R=
EffectOfFatigue OnProductivity f(Fatigue P3R)
~ dmnl
~ |

Fatigue effect PDY P3T=
EffectOfFatigue OnProductivity f(Fatigue P3T)
~ dmnl
~ |

Fatigue effect PDY P4D=
EffectOfFatigue OnProductivity f(Fatigue P4D)
~ dmnl
~ |

Fatigue effect PDY P4H=
EffectOfFatigue OnProductivity f(Fatigue P4H)

~ dmnl
~ |

Fatigue effect PDY P4R=
EffectOfFatigue OnProductivity f(Fatigue P4R)
~ dmnl
~ |

Fatigue effect PDY P4T=
EffectOfFatigue OnProductivity f(Fatigue P4T)
~ dmnl
~ |

Fatigue effect qual P1D=
EffectOfFatigue OnQuality f(Fatigue P1D)
~ dmnl
~ |

Fatigue effect qual P1H=
EffectOfFatigue OnQuality f(Fatigue P1H)
~ dmnl
~ |

Fatigue effect qual P1R=
EffectOfFatigue OnQuality f(Fatigue P1R)
~ dmnl
~ |

Fatigue effect qual P1T=
EffectOfFatigue OnQuality f(Fatigue P1T)
~ dmnl
~ |

Fatigue effect qual P2D=
EffectOfFatigue OnQuality f(Fatigue P2D)
~ dmnl
~ |

Fatigue effect qual P2H=
EffectOfFatigue OnQuality f(Fatigue P2H)
~ dmnl
~ |

Fatigue effect qual P2R=
EffectOfFatigue OnQuality f(Fatigue P2R)
~ dmnl
~ |

Fatigue effect qual P2T=
EffectOfFatigue OnQuality f(Fatigue P2T)
~ dmnl
~ |

Fatigue effect qual P3D=
EffectOfFatigue OnQuality f(Fatigue P3D)
~ dmnl
~ |

Fatigue effect qual P3H=
EffectOfFatigue OnQuality f(Fatigue P3H)
~ dmnl
~ |

Fatigue effect qual P3R=
EffectOfFatigue OnQuality f(Fatigue P3R)
~ dmnl
~ |

Fatigue effect qual P3T=
EffectOfFatigue OnQuality f(Fatigue P3T)
~ dmnl
~ |

```
Fatigue effect qual P4D=
	EffectOfFatigue OnQuality f(Fatigue P4D)
	~	dmnl
	~		|

Fatigue effect qual P4H=
	EffectOfFatigue OnQuality f(Fatigue P4H)
	~	dmnl
	~		|

Fatigue effect qual P4R=
	EffectOfFatigue OnQuality f(Fatigue P4R)
	~	dmnl
	~		|

Fatigue effect qual P4T=
	EffectOfFatigue OnQuality f(Fatigue P4T)
	~	dmnl
	~		|

Fatigue P1D=
	SMOOTHI(OverTime P1D,TimeToGetFatigued D,1)
	~	fraction
	~		|

Fatigue P1H=
	SMOOTHI(OverTime P1H,TimeToGetFatigued H,1)
	~	fraction
	~		|

Fatigue P1R=
	SMOOTHI(OverTime P1R,TimeToGetFatigued R,1)
	~	fraction
	~		|

Fatigue P1T=
	SMOOTHI(OverTime P1T,TimeToGetFatigued T,1)
	~	fraction
	~		|

FindBugs P1D = HiddenBugs P1D/BugFindTime D
	~	lines/Month
	~		|

FindBugs P1H = HiddenBugs P1H/BugFindTime H
	~	lines/Month
	~		|

FindBugs P1R = HiddenBugs P1R/BugFindTime R
	~	lines/Month
	~		|

FindBugs P1T = HiddenBugs P1T/BugFindTime T
	~	lines/Month
	~		|

Gap Effect on Hiring D f(
	[(0,0)-(1e+006,20)],(0,2),(1e-005,2),(0.25,2),(0.5,1.5),(0.9,1),(1,0.9),(1.1,0.8),(2\
		,0.7),(10,0.2),(100,0.1),(1e+006,0.01))
	~	fraction
	~		|

Gap Effect on Hiring H f(
	[(0,0)-(1e+009,20)],(0,2),(1e-005,2),(0.25,2),(0.5,1.5),(0.9,1),(1,0.9),(1.1,0.8),(2\
		,0.7),(10,0.2),(100,0.1),(1e+009,0.01))
	~	fraction
	~		|

Gap Effect on Hiring R f(
	[(0,0)-(1e+009,20)],(0,2),(1e-005,2),(0.25,2),(0.5,1.5),(0.9,1),(1,0.9),(1.1,0.8),(2\
		,0.7),(10,0.2),(100,0.1),(1e+009,0.01))
	~	fraction
	~		|
```

GapP1ED=
AllocatedP1ED-P1ED
~ people
~ |

GapP1EH=
AllocatedP1EH-P1EH
~ people
~ |

GapP1ER=
AllocatedP1ER-P1ER
~ people
~ |

GapP1ET=
AllocatedP1ET-P1ET
~ people
~ |

GapP1ID=
AllocatedP1ID-P1ID
~ people
~ |

GapP1IH=
AllocatedP1IH-P1IH
~ people
~ |

GapP1IR=
AllocatedP1IR-P1IR
~ people
~ |

GapP1IT=
AllocatedP1IT-P1IT
~ people
~ |

GapP1ND=
AllocatedP1ND-P1ND
~ people
~ |

GapP1NH=
AllocatedP1NH-P1NH
~ people
~ |

GapP1NR=
AllocatedP1NR-P1NR
~ people
~ |

GapP1NT=
AllocatedP1NT-P1NT
~ people
~ |

GapP2ED=
AllocatedP2ED-P2ED
~ people
~ |

GapP2EH=
AllocatedP2EH-P2EH
~ people
~ |

GapP2ER=
AllocatedP2ER-P2ER

```
	~	people
	~		|

GapP2ET=
	AllocatedP2ET-P2ET
	~	people
	~		|

GapP2ID=
	AllocatedP2ID-P2ID
	~	people
	~		|

GapP2IH=
	AllocatedP2IH-P2IH
	~	people
	~		|

GapP2IT=
	AllocatedP2IT-P2IT
	~	people
	~		|

GapP2ND=
	AllocatedP2ND-P2ND
	~	people
	~		|

GapP2NH=
	AllocatedP2NH-P2NH
	~	people
	~		|

GapP2NR=
	AllocatedP2NR-P2NR
	~	people
	~		|

GapP2NT=
	AllocatedP2NT-P2NT
	~	people
	~		|

GapP3ED=
	AllocatedP3ED-P3ED
	~	people
	~		|

GapP3EH=
	AllocatedP3EH-P3EH
	~	people
	~		|

GapP3ER=
	AllocatedP3ER-P3ER
	~	people
	~		|

GapP3ET=
	AllocatedP3ET-P3ET
	~	people
	~		|

GapP3ID=
	AllocatedP3ID-P3ID
	~	people
	~		|

GapP3IH=
	AllocatedP3IH-P3IH
	~	people
	~		|
```

GapP3IR=
AllocatedP3IR-P3IR
~ people
~ |

GapP3IT=
AllocatedP3IT-P3IT
~ people
~ |

GapP3ND=
AllocatedP3ND-P3ND
~ people
~ |

GapP3NH=
AllocatedP3NH-P3NH
~ people
~ |

GapP3NR=
AllocatedP3NR-P3NR
~ people
~ |

GapP3NT=
AllocatedP3NT-P3NT
~ people
~ |

GapP4ED=
AllocatedP4ED-P4ED
~ people
~ |

GapP4EH=
AllocatedP4EH-P4EH
~ people
~ |

GapP4ER=
AllocatedP4ER-P4ER
~ people
~ |

GapP4ET=
AllocatedP4ET-P4ET
~ people
~ |

GapP4ID=
AllocatedP4ID-P4ID
~ people
~ |

GapP4IH=
AllocatedP4IR 0-P4IH
~ people
~ |

GapP4IR=
AllocatedP4IR-P4IR
~ people
~ |

GapP4IT=
AllocatedP4IT-P4IT
~ people
~ |

GapP4ND=
AllocatedP4ND-P4ND
~ people

~ |

GapP4NH=
AllocatedP4NH-P4NH
~ people
~ |

GapP4NR=
AllocatedP4NR-P4NR
~ people
~ |

GapP4NT=
AllocatedP4NT-P4NT
~ people
~ |

GapRatio D=
ZIDZ((TotalND+TotalID+TotalED),(SUM(NDDesired[project!])+SUM(IDDesired[project!])+SUM\
(EDDesired[project!])))
~ fraction
~ |

GapRatio H=
ZIDZ((TotalNH+TotalIH+TotalEH),(SUM(NHDesired[project!])+SUM(IHDesired[project!])+SUM\
(EHDesired[project!])))
~ fraction
~ |

GapRatio R=
ZIDZ((TotalNR+TotalIR+TotalER),(SUM(NRDesired[project!])+SUM(IRDesired[project!])+SUM\
(ERDesired[project!])))
~ fraction
~ |

HiddenBugs P1D = INTEG(Doing wrong P1D - FindBugs P1D, 0)
~ lines
~ |

HiddenBugs P1H = INTEG(Doing wrong P1H - FindBugs P1H, 0)
~ lines
~ |

HiddenBugs P1R = INTEG(Doing wrong P1R - FindBugs P1R, 0)
~ lines
~ |

HiddenBugs P1T = INTEG(Doing wrong P1T - FindBugs P1T, 0)
~ lines
~ |

HiddenBugs P2D = INTEG(Doing wrong P2D - FindBugs P2D, 0)
~ lines
~ |

HiddenBugs P2H = INTEG(Doing wrong P2H - FindBugs P2H, 0)
~ lines
~ |

HiddenBugs P2R = INTEG(Doing wrong P2R - FindBugs P2R, 0)
~ lines
~ |

HiddenBugs P2T = INTEG(Doing wrong P2T - FindBugs P2T, 0)
~ lines
~ |

HiddenBugs P3D = INTEG(Doing wrong P3D - FindBugs P3D, 0)
~ lines
~ |

HiddenBugs P3H = INTEG(Doing wrong P3H - FindBugs P3H, 0)
~ lines

```
	~		|

HiddenBugs P3R = INTEG(Doing wrong P3R - FindBugs P3R, 0)
	~	lines
	~		|

HiddenBugs P3T = INTEG(Doing wrong P3T - FindBugs P3T, 0)
	~	lines
	~		|

HiddenBugs P4D = INTEG(Doing wrong P4D - FindBugs P4D, 0)
	~	lines
	~		|

HiddenBugs P4H = INTEG(Doing wrong P4H - FindBugs P4H, 0)
	~	lines
	~		|

HiddenBugs P4R = INTEG(Doing wrong P4R - FindBugs P4R, 0)
	~	lines
	~		|

HiddenBugs P4T = INTEG(Doing wrong P4T - FindBugs P4T, 0)
	~	lines
	~		|

HLD Start Rate P1H=
	if then else(Initial WorkToDo P1H>0,(Doing right P1R/InitialWorkToDo P1R)*Init HLD P1H\
		,0)
	~	lines/Month
	~		|

HLD Start Rate P2H=
	if then else(Initial WorkToDo P2H>0,(Doing right P2R/InitialWorkToDo P2R)*Init HLD P2H\
		,0)
	~	lines/Month
	~		|

HLD Start Rate P3H=
	if then else(Initial WorkToDo P3H>0,(Doing right P3R/InitialWorkToDo P3R)*Init HLD P3H\
		,0)
	~	lines/Month
	~		|

HLD Start Rate P4H=
	if then else(Initial WorkToDo P4H>0,(Doing right P4R/InitialWorkToDo P4R)*Init HLD P4H\
		,0)
	~	lines/Month
	~		|

IDAllocated[project]=
	Allocate By Priority(IDDesired[project],IDAttractiveness[project],4,1,TotalID)
	~	people
	~		|

IDAttractiveness[one]=
	Attractiveness P1D ~~|
IDAttractiveness[two]=
	Attractiveness P2D ~~|
IDAttractiveness[three]=
	Attractiveness P3D ~~|
IDAttractiveness[four]=
	Attractiveness P4D
	~	dmnl
	~		|

IDDesired[one]=
	DesiredP1ID ~~|
IDDesired[two]=
	DesiredP2ID ~~|
IDDesired[three]=
	DesiredP3ID ~~|
IDDesired[four]=
```

DesiredP4ID
~ people
~ |

IHAllocated[project]=
Allocate By Priority(IHDesired[project],IHAttractiveness[project],4,1,TotalIH)
~ people
~ |

IHAttractiveness[one]=
Attractiveness P1H ~~|
IHAttractiveness[two]=
Attractiveness P2H ~~|
IHAttractiveness[three]=
Attractiveness P3H ~~|
IHAttractiveness[four]=
Attractiveness P4H
~ dmnl
~ |

IHDesired[one]=
DesiredP1IH ~~|
IHDesired[two]=
DesiredP2IH ~~|
IHDesired[three]=
DesiredP3IH ~~|
IHDesired[four]=
DesiredP4IH
~ people
~ |

Indicated overtime P1D=
DesiredPeople P1D/Workforce P1D
~ fraction
~ |

Indicated overtime P1H=
DesiredPeople P1H/Workforce P1H
~ fraction
~ |

Indicated overtime P1R=
DesiredPeople P1R/(Workforce P1R)
~ fraction
~ |

Indicated overtime P1T=
DesiredPeople P1T/Workforce P1T
~ fraction
~ |

Indicated overtime P2D=
DesiredPeople P2D/Workforce P2D
~ fraction
~ |

Indicated overtime P2H=
DesiredPeople P2H/Workforce P2H
~ fraction
~ |

Indicated overtime P2R=
DesiredPeople P2R/Workforce P2R
~ fraction
~ |

Indicated overtime P2T=
DesiredPeople P2T/Workforce P2T
~ fraction
~ |

Indicated overtime P3D=
DesiredPeople P3D/Workforce P3D

```
	~	fraction
	~		|

Indicated overtime P3H=
	DesiredPeople P3H/Workforce P3H
	~	fraction
	~		|

Indicated overtime P3R=
	DesiredPeople P3R/Workforce P3R
	~	fraction
	~		|

Indicated overtime P3T=
	DesiredPeople P3T/Workforce P3T
	~	fraction
	~		|

Indicated overtime P4D=
	DesiredPeople P4D/Workforce P4D
	~	fraction
	~		|

Indicated overtime P4H=
	DesiredPeople P4H/Workforce P4H
	~	fraction
	~		|

Indicated overtime P4R=
	DesiredPeople P4R/Workforce P4R
	~	fraction
	~		|

Indicated overtime P4T=
	DesiredPeople P4T/Workforce P4T
	~	fraction
	~		|

Init Dev P1D=
	Get XLS Constants('ModelConstants.xls','Project Constants','q2')
	~	lines
	~		|

Init Dev P2D=
	Get XLS Constants('ModelConstants.xls','Project Constants','q3')
	~	lines
	~		|

Init Dev P3D=
	Get XLS Constants('ModelConstants.xls','Project Constants','q4')
	~	lines
	~		|

Init Dev P4D=
	Get XLS Constants('ModelConstants.xls','Project Constants','q5')
	~	lines
	~		|

Init HLD P1H=
	Get XLS Constants('ModelConstants.xls','Project Constants','p2')
	~	lines
	~		|

Init HLD P2H=
	Get XLS Constants('ModelConstants.xls','Project Constants','p3')
	~	lines
	~		|

Init HLD P3H=
	Get XLS Constants('ModelConstants.xls','Project Constants','p4')
	~	lines
	~		|
```

```
Init HLD P4H=
	Get XLS Constants('ModelConstants.xls','Project Constants','p5')
	~	lines
	~		|

Init Test P1T=
	Get XLS Constants('ModelConstants.xls','Project Constants','r2')
	~	lines
	~		|

Init Test P2T=
	Get XLS Constants('ModelConstants.xls','Project Constants','r3')
	~	lines
	~		|

Init Test P3T=
	Get XLS Constants('ModelConstants.xls','Project Constants','r4')
	~	lines
	~		|

Init Test P4T=
	Get XLS Constants('ModelConstants.xls','Project Constants','r5')
	~	lines
	~		|

Initial WorkToDo P1D= INTEG (
	-Start Dev Rate P1D,
		Init Dev P1D)
	~	lines
	~		|

Initial WorkToDo P1H= INTEG (
	-HLD Start Rate P1H,
		Init HLD P1H)
	~	lines
	~		|

Initial WorkToDo P1T= INTEG (
	-Test Start Rate P1T,
		Init Test P1T)
	~	lines
	~		|

Initial WorkToDo P2D= INTEG (
	-Start Dev Rate P2D,
		Init Dev P2D)
	~	lines
	~		|

Initial WorkToDo P2H= INTEG (
	-HLD Start Rate P2H,
		Init HLD P2H)
	~	lines
	~		|

Initial WorkToDo P2T= INTEG (
	-Test Start Rate P2T,
		Init Test P2T)
	~	lines
	~		|

Initial WorkToDo P3D= INTEG (
	-Start Dev Rate P3D,
		Init Dev P3D)
	~	lines
	~		|

Initial WorkToDo P3H= INTEG (
	-HLD Start Rate P3H,
		Init HLD P3H)
	~	lines
	~		|
```

```
Initial WorkToDo P3T= INTEG (
        -Test Start Rate P3T,
                Init Test P3T)
        ~       lines
        ~               |

Initial WorkToDo P4D= INTEG (
        -Start Dev Rate P4D,
                Init Dev P4D)
        ~       lines
        ~               |

Initial WorkToDo P4H= INTEG (
        -HLD Start Rate P4H,
                Init HLD P4H)
        ~       lines
        ~               |

Initial WorkToDo P4T= INTEG (
        -Test Start Rate P4T,
                Init Test P4T)
        ~       lines
        ~               |

InitialDueDate P1D=
        Get XLS Constants('ModelConstants.xls','Project Constants','u2')
        ~       Month
        ~               |

InitialDueDate P1H=
        Get XLS Constants('ModelConstants.xls','Project Constants','t2')
        ~       Month
        ~               |

InitialDueDate P1R=
        Get XLS Constants('ModelConstants.xls','Project Constants','s2')
        ~       Month
        ~               |

InitialDueDate P1T=
        Get XLS Constants('ModelConstants.xls','Project Constants','v2')
        ~       Month
        ~               |

InitialDueDate P2D=
        Get XLS Constants('ModelConstants.xls','Project Constants','u3')
        ~       Month
        ~               |

InitialDueDate P2H=
        Get XLS Constants('ModelConstants.xls','Project Constants','t3')
        ~       Month
        ~               |

InitialDueDate P2R=
        Get XLS Constants('ModelConstants.xls','Project Constants','s3')
        ~       Month
        ~               |

InitialDueDate P2T=
        Get XLS Constants('ModelConstants.xls','Project Constants','v3')
        ~       Month
        ~               |

InitialDueDate P3D=
        Get XLS Constants('ModelConstants.xls','Project Constants','u4')
        ~       Month
        ~               |

InitialDueDate P3H=
        Get XLS Constants('ModelConstants.xls','Project Constants','t4')
        ~       Month
        ~               |
```

InitialDueDate P3R=
Get XLS Constants('ModelConstants.xls','Project Constants','s4')
~ Month
~ |

InitialDueDate P3T=
Get XLS Constants('ModelConstants.xls','Project Constants','v4')
~ Month
~ |

InitialDueDate P4D=
Get XLS Constants('ModelConstants.xls','Project Constants','u5')
~ Month
~ |

InitialDueDate P4H=
Get XLS Constants('ModelConstants.xls','Project Constants','t5')
~ Month
~ |

InitialDueDate P4R=
Get XLS Constants('ModelConstants.xls','Project Constants','s5')
~ Month
~ |

InitialDueDate P4T=
Get XLS Constants('ModelConstants.xls','Project Constants','v5')
~ Month
~ |

InitialWorkToDo P1R=
Get XLS Constants('ModelConstants.xls','Project Constants','o2')
~ lines
~ |

InitialWorkToDo P1R Pulse=
InitialWorkToDo P1R*pulse(TimeToStart P1R,TIME STEP)*(1/TIME STEP)
~ lines/Month
~ |

InitialWorkToDo P2R=
Get XLS Constants('ModelConstants.xls','Project Constants','o3')
~ lines
~ |

InitialWorkToDo P2R Pulse=
InitialWorkToDo P2R*pulse(TimeToStart P2R,TIME STEP)*(1/TIME STEP)
~ lines/Month
~ |

InitialWorkToDo P3R=
Get XLS Constants('ModelConstants.xls','Project Constants','o4')
~ lines
~ |

InitialWorkToDo P3R Pulse=
InitialWorkToDo P3R*pulse(TimeToStart P3R,TIME STEP)*(1/TIME STEP)
~ lines/Month
~ |

InitialWorkToDo P4R=
Get XLS Constants('ModelConstants.xls','Project Constants','o5')
~ lines
~ |

InitialWorkToDo P4R Pulse=
InitialWorkToDo P4R*pulse(TimeToStart P4R,TIME STEP)*(1/TIME STEP)
~ lines/Month
~ |

Intermediate Advance to Expert Time R=
Get XLS Constants('ModelConstants.xls','Phase Constants','b3')

```
	~	Month
	~		|

Intermediate Attrition=
	0.1
	~	fraction
	~	10 percent / year
	|

Intermediate Skill Effect on Quality=
	Get XLS Constants('ModelConstants.xls','Portfolio Constants','b21')
	~	fraction
	~		|

IRAllocated[project]=
	Allocate By Priority(IRDesired[project],IRAttractiveness[project],4,1,TotalIR)
	~	people
	~		|

IRAttractiveness[one]=
	Attractiveness P1R ~~|
IRAttractiveness[two]=
	Attractiveness P2R ~~|
IRAttractiveness[three]=
	Attractiveness P3R ~~|
IRAttractiveness[four]=
	Attractiveness P4R
	~	dmnl
	~		|

IRDesired[one]=
	DesiredP1IR ~~|
IRDesired[two]=
	DesiredP2IR ~~|
IRDesired[three]=
	DesiredP3IR ~~|
IRDesired[four]=
	DesiredP4IR
	~	people
	~		|

ITAllocated[project]=
	Allocate By Priority(ITDesired[project],ITAttractiveness[project],4,1,TotalIT)
	~	people
	~		|

ITAttractiveness[one]=
	Attractiveness P1T ~~|
ITAttractiveness[two]=
	Attractiveness P2T ~~|
ITAttractiveness[three]=
	Attractiveness P3T ~~|
ITAttractiveness[four]=
	Attractiveness P4T
	~	dmnl
	~		|

ITDesired[one]=
	DesiredP1IT ~~|
ITDesired[two]=
	DesiredP2IT ~~|
ITDesired[three]=
	DesiredP3IT ~~|
ITDesired[four]=
	DesiredP4IT
	~	people
	~		|

MaxQuality D=
	Get XLS Constants('ModelConstants.xls','Phase Constants','d7')
	~	fraction
	~	For full model, change to 0.6
	|
```

```
MaxQuality H=
	Get XLS Constants('ModelConstants.xls','Phase Constants','c7')
	~	fraction
	~	For full model, change to 0.6
	|

MaxQuality R=
	Get XLS Constants('ModelConstants.xls','Phase Constants','b7')
	~	fraction
	~	For full model, change to 0.6
	|

MaxQuality T=
	Get XLS Constants('ModelConstants.xls','Phase Constants','e7')
	~	fraction
	~	For full model, change to 0.6
	|

minimum remaining time D=
	Get XLS Constants('ModelConstants.xls','Phase Constants','d4')
	~	months
	~		|

minimum remaining time H=
	Get XLS Constants('ModelConstants.xls','Phase Constants','c4')
	~	Month
	~		|

minimum remaining time R=
	Get XLS Constants('ModelConstants.xls','Phase Constants','b4')
	~	Month
	~		|

minimum remaining time T=
	Get XLS Constants('ModelConstants.xls','Phase Constants','e4')
	~	Month
	~		|

NDAllocated[project]=
	Allocate By Priority(NDDesired[project],NDAttractiveness[project],4,1,TotalND)
	~	people
	~		|

NDAttractiveness[one]=
	Attractiveness P1D ~~|
NDAttractiveness[two]=
	Attractiveness P2D ~~|
NDAttractiveness[three]=
	Attractiveness P3D ~~|
NDAttractiveness[four]=
	Attractiveness P4D
	~	dmnl
	~		|

NDDesired[one]=
	DesiredP1ND ~~|
NDDesired[two]=
	DesiredP2ND ~~|
NDDesired[three]=
	DesiredP3ND ~~|
NDDesired[four]=
	DesiredP4ND
	~	people
	~		|

NHAllocated[project]=
	Allocate By Priority(NHDesired[project],NHAttractiveness[project],4,1,TotalNH)
	~	people
	~		|

NHAttractiveness[one]=
	Attractiveness P1H ~~|
```

```
NHAttractiveness[two]=
	Attractiveness P2H ~~|
NHAttractiveness[three]=
	Attractiveness P3H ~~|
NHAttractiveness[four]=
	Attractiveness P4H
	~	dmnl
	~		|

NHDesired[one]=
	DesiredP1NH ~~|
NHDesired[two]=
	DesiredP2NH ~~|
NHDesired[three]=
	DesiredP3NH ~~|
NHDesired[four]=
	DesiredP4NH
	~	people
	~		|

Normal Productivity D=
	Get XLS Constants('ModelConstants.xls','Phase Constants','d8')
	~	lines/(people*Month)
	~		|

Normal Productivity H=
	Get XLS Constants('ModelConstants.xls','Phase Constants','c8')
	~	lines/(people*Month)
	~		|

Normal Productivity R=
	Get XLS Constants('ModelConstants.xls','Phase Constants','b8')
	~	lines/(people*Month)
	~		|

Normal Productivity T=
	Get XLS Constants('ModelConstants.xls','Phase Constants','e8')
	~	lines/(people*Month)
	~		|

Novice Advance to Intermediate Time R=
	Get XLS Constants('ModelConstants.xls','Phase Constants','b2')
	~	Month
	~		|

Novice Attrition=
	0.1
	~	fraction
	~	10 percent / year
	|

Novice Skill Effect on Quality=
	Get XLS Constants('ModelConstants.xls','Portfolio Constants','b20')
	~	fraction
	~		|

NRAllocated[project]=
	Allocate By Priority(NRDesired[project],NRAttractiveness[project],4,1,TotalNR)
	~	people
	~		|

NRAttractiveness[one]=
	Attractiveness P1R ~~|
NRAttractiveness[two]=
	Attractiveness P2R ~~|
NRAttractiveness[three]=
	Attractiveness P3R ~~|
NRAttractiveness[four]=
	Attractiveness P4R
	~	dmnl
	~		|

NRDesired[one]=
```

```
	DesiredP1NR ~~|
NRDesired[two]=
	DesiredP2NR ~~|
NRDesired[three]=
	DesiredP3NR ~~|
NRDesired[four]=
	DesiredP4NR
	~	people
	~		|

NTAllocated[project]=
	Allocate By Priority(NTDesired[project],NTAttractiveness[project],4,1,TotalNT)
	~	people
	~		|

NTAttractiveness[one]=
	Attractiveness P1T ~~|
NTAttractiveness[two]=
	Attractiveness P2T ~~|
NTAttractiveness[three]=
	Attractiveness P3T ~~|
NTAttractiveness[four]=
	Attractiveness P4T
	~	dmnl
	~		|

NTDesired[one]=
	DesiredP1NT ~~|
NTDesired[two]=
	DesiredP2NT ~~|
NTDesired[three]=
	DesiredP3NT ~~|
NTDesired[four]=
	DesiredP4NT
	~	people
	~		|

OverTime f(
	[(0,0)-(3.40282e+038,2)],(0,0),(1,1),(1.81269,1),(2.5,1),(10,1.5),(1e+030,1.5))
	~	fraction
	~	[(0,0)-(2.5,2)],(0,0),(1,1),(1.81269,1.60526),(2.5,2)\!\!\!
	|

OverTime P1D=
	OverTime f(Indicated overtime P1D)
	~	fraction
	~		|

OverTime P1H=
	OverTime f(Indicated overtime P1H)
	~	fraction
	~		|

OverTime P1R=
	OverTime f(Indicated overtime P1R)
	~	fraction
	~		|

OverTime P1T=
	OverTime f(Indicated overtime P1T)
	~	fraction
	~		|

OverTime P2D=
	OverTime f(Indicated overtime P2D)
	~	fraction
	~		|

OverTime P2H=
	OverTime f(Indicated overtime P2H)
	~	fraction
	~		|
```

```
OverTime P2R=
	OverTime f(Indicated overtime P2R)
	~	fraction
	~		|

OverTime P2T=
	OverTime f(Indicated overtime P2T)
	~	fraction
	~		|

OverTime P3D=
	OverTime f(Indicated overtime P3D)
	~	fraction
	~		|

OverTime P3H=
	OverTime f(Indicated overtime P3H)
	~	fraction
	~		|

OverTime P3R=
	OverTime f(Indicated overtime P3R)
	~	fraction
	~		|

OverTime P3T=
	OverTime f(Indicated overtime P3T)
	~	fraction
	~		|

OverTime P4D=
	OverTime f(Indicated overtime P4D)
	~	fraction
	~		|

OverTime P4H=
	OverTime f(Indicated overtime P4H)
	~	fraction
	~		|

OverTime P4R=
	OverTime f(Indicated overtime P4R)
	~	fraction
	~		|

OverTime P4T=
	OverTime f(Indicated overtime P4T)
	~	fraction
	~		|

P1 Initial Priority=
	Get XLS Constants('ModelConstants.xls','Project Constants','b2')
	~	dmnl
	~		|

P1DDesiredE=
	DesiredRealHeads P1D*RatioED
	~	people
	~		|

P1DDesiredI=
	DesiredRealHeads P1D*RatioID
	~	people
	~		|

P1DDesiredN=
	DesiredRealHeads P1D*RatioND
	~	people
	~		|

P1ED= INTEG (
	P1ED Rate+P1IDtoED Rate-Attrition Rate P1ED,
		StartP1ED)
```

~ people
~ |

P1EH= INTEG (
P1EH Rate+P1IHtoEH Rate-Attrition Rate P1EH,
StartP1EH)
~ people
~ |

P1ER= INTEG (
P1ER Rate+P1IRtoER Rate-Attrition Rate P1ER,
StartP1ER)
~ people
~ |

P1HDesiredE=
DesiredRealHeads P1H*RatioEH
~ people
~ |

P1HDesiredI=
DesiredRealHeads P1H*RatioIH
~ people
~ |

P1HDesiredN=
DesiredRealHeads P1H*RatioNH
~ people
~ |

P1ID= INTEG (
P1ID Rate+P1NDtoID Rate-P1IDtoED Rate-Attrition Rate P1ID,
StartP1ID)
~ people
~ |

P1IDtoED Rate=
(P1ID/Intermediate Advance to Expert Time D)*Staffing Gap effect on learning P1D*Complexity effect on learning P1
~ people/Month
~ |

P1IH= INTEG (
P1IH Rate+P1NHtoIH Rate-P1IHtoEH Rate-Attrition Rate P1IH,
StartP1IH)
~ people
~ |

P1IHtoEH Rate=
(P1IH/Intermediate Advance to Expert Time H)*Staffing Gap effect on learning P1H*Complexity effect on learning P1
~ people/Month
~ |

P1IR= INTEG (
P1IR Rate+P1NRtoIR Rate-P1IRtoER Rate-Attrition Rate P1IR,
StartP1IR)
~ people
~ |

P1IRtoER Rate=
(P1IR/Intermediate Advance to Expert Time R)*Staffing Gap effect on learning P1R*Complexity effect on learning P1
~ people/Month
~ |

P1ITtoET Rate=
(P1IT/Intermediate Advance to Expert Time T)*Staffing Gap effect on learning P1T*Complexity effect on learning P1
~ people/Month
~ |

P1ND= INTEG (
P1ND Rate-P1NDtoID Rate-Attrition Rate P1ND,
StartP1ND)
~ people
~ |

P1NDtoID Rate=
(P1ND/Novice Advance to Intermediate Time D)*Staffing Gap effect on learning P1D*Complexity effect on learning P1
~ people/Month
~ |

P1NH= INTEG (
P1NH Rate-P1NHtoIH Rate-Attrition Rate P1NH,
StartP1NH)
~ people
~ |

P1NHtoIH Rate=
(P1NH/Novice Advance to Intermediate Time H)*Staffing Gap effect on learning P1H*Complexity effect on learning P1
~ people/Month
~ |

P1NR= INTEG (
P1NR Rate-P1NRtoIR Rate-Attrition Rate P1NR,
StartP1NR)
~ people
~ |

P1NRtoIR Rate=
(P1NR/Novice Advance to Intermediate Time R)*Staffing Gap effect on learning P1R*Complexity effect on learning P1
~ people/Month
~ |

P1NTtoIT Rate=
(P1NT/Novice Advance to Intermediate Time T)*Staffing Gap effect on learning P1T*Complexity effect on learning P1
~ people/Month
~ |

P1TDesiredE=
DesiredRealHeads P1T*RatioET
~ people
~ |

P1TDesiredI=
DesiredRealHeads P1T*RatioIT
~ people
~ |

P1TDesiredN=
DesiredRealHeads P1T*RatioNT
~ people
~ |

P2 Initial Priority=
Get XLS Constants('ModelConstants.xls','Project Constants','b3')
~ dmnl
~ |

P2DDesiredE=
DesiredRealHeads P2D*RatioED
~ people
~ |

P2DDesiredI=
DesiredRealHeads P2D*RatioID
~ people
~ |

P2DDesiredN=
DesiredRealHeads P2D*RatioND
~ people
~ |

P2ED= INTEG (
P2ED Rate+P2IDtoED Rate-Attrition Rate P2ED,
StartP2ED)
~ people
~ |

P2ED Rate=
if then else(GapP2ED>0,Min(GapP2ED, ED Control)/TimeToMoveIn,GapP2ED/TimeToMoveOut)
~ people/Month
~ |

P2EH= INTEG (
P2EH Rate+P2IHtoEH Rate-Attrition Rate P2EH,
StartP2EH)
~ people
~ |

P2EH Rate=
if then else(GapP2EH>0,Min(GapP2EH, EH Control)/TimeToMoveIn,GapP2EH/TimeToMoveOut)
~ people/Month
~ |

P2ER= INTEG (
P2ER Rate+P2IRtoER Rate-Attrition Rate P2ER,
StartP2ER)
~ people
~ |

P2ER Rate=
if then else(GapP2ER>0,Min(GapP2ER, ER Control)/TimeToMoveIn,GapP2ER/TimeToMoveOut)
~ people/Month
~ |

P2ET Rate=
if then else(GapP2ET>0,Min(GapP2ET, ET Control)/TimeToMoveIn,GapP2ET/TimeToMoveOut)
~ people/Month
~ |

P2HDesiredE=
DesiredRealHeads P2H*RatioEH
~ people
~ |

P2HDesiredI=
DesiredRealHeads P2H*RatioIH
~ people
~ |

P2HDesiredN=
DesiredRealHeads P2H*RatioNH
~ people
~ |

P2ID= INTEG (
P2ID Rate+P2NDtoID Rate-P2IDtoED Rate-Attrition Rate P2ID,
StartP2ID)
~ people
~ |

P2ID Rate=
if then else(GapP2ID>0,Min(GapP2ID, ID Control)/TimeToMoveIn,GapP2ID/TimeToMoveOut)
~ people/Month
~ |

P2IH= INTEG (
P2IH Rate+P2NHtoIH Rate-P2IHtoEH Rate-Attrition Rate P2IH,
StartP2IH)
~ people
~ |

P2IH Rate=
if then else(GapP2IH>0,Min(GapP2IH,IH Control)/TimeToMoveIn,GapP2IH/TimeToMoveOut)
~ people/Month
~ |

P2IR= INTEG (
P2IR Rate+P2NRtoIR Rate-P2IRtoER Rate-Attrition Rate P2IR,
StartP2IR)

~ people
~ |

P2IR Rate=
if then else(GapP2IR>0,Min(GapP2IR, IR Control)/TimeToMoveIn,GapP2IR/TimeToMoveOut)
~ people/Month
~ |

P2IT Rate=
if then else(GapP2IT>0,Min(GapP2IT, IT Control)/TimeToMoveIn,GapP2IT/TimeToMoveOut)
~ people/Month
~ |

P2ND= INTEG (
P2ND Rate-P2NDtoID Rate-Attrition Rate P2ND,
StartP2ND)
~ people
~ |

P2ND Rate=
if then else(GapP2ND>0,Min(GapP2ND, ND Control)/TimeToMoveIn,GapP2ND/TimeToMoveOut)
~ people/Month
~ |

P2NDtoID Rate=
(P2ND/Novice Advance to Intermediate Time D)*Staffing Gap effect on learning P2D*Complexity effect on learning P2
~ people/Month
~ |

P2NH= INTEG (
P2NH Rate-P2NHtoIH Rate-Attrition Rate P2NH,
StartP2NH)
~ people
~ |

P2NH Rate=
if then else(GapP2NH>0,Min(GapP2NH, NH Control)/TimeToMoveIn,GapP2NH/TimeToMoveOut)
~ people/Month
~ |

P2NHtoIH Rate=
(P2NH/Novice Advance to Intermediate Time H)*Staffing Gap effect on learning P2H*Complexity effect on learning P2
~ people/Month
~ |

P2NR= INTEG (
P2NR Rate-P2NRtoIR Rate-Attrition Rate P2NR,
StartP2NR)
~ people
~ |

P2NR Rate=
if then else(GapP2NR>0,Min(GapP2NR, NR Control)/TimeToMoveIn,GapP2NR/TimeToMoveOut)
~ people/Month
~ |

P2NRtoIR Rate=
(P2NR/Novice Advance to Intermediate Time R)*Staffing Gap effect on learning P2R*Complexity effect on learning P2
~ people/Month
~ |

P2NT Rate=
if then else(GapP2NT>0,Min(GapP2NT, NT Control)/TimeToMoveIn,GapP2NT/TimeToMoveOut)
~ people/Month
~ |

P2NTtoIT Rate=
(P2NT/Novice Advance to Intermediate Time T)*Staffing Gap effect on learning P2T*Complexity effect on learning P2
~ people/Month
~ |

P2RDesiredE=
DesiredRealHeads P2R*RatioER

~ people
~ |

P2RDesiredI=
DesiredRealHeads P2R*RatioIR
~ people
~ |

P2RDesiredN=
DesiredRealHeads P2R*RatioNR
~ people
~ |

P2TDesiredE=
DesiredRealHeads P2T*RatioET
~ people
~ |

P2TDesiredI=
DesiredRealHeads P2T*RatioIT
~ people
~ |

P2TDesiredN=
DesiredRealHeads P2T*RatioNT
~ people
~ |

P3 Initial Priority=
Get XLS Constants('ModelConstants.xls','Project Constants','b4')
~ dmnl
~ |

P3DDesiredE=
DesiredRealHeads P3D*RatioED
~ people
~ |

P3DDesiredI=
DesiredRealHeads P3D*RatioID
~ people
~ |

P3DDesiredN=
DesiredRealHeads P3D*RatioND
~ people
~ |

P3ED= INTEG (
P3ED Rate+P3IDtoED Rate-Attrition Rate P3ED,
StartP3ED)
~ people
~ |

P3ED Rate=
if then else(GapP3ED>0,Min(GapP3ED, ED Control)/TimeToMoveIn,GapP3ED/TimeToMoveOut)
~ people/Month
~ |

P3EH= INTEG (
P3EH Rate+P3IHtoEH Rate-Attrition Rate P3EH,
StartP3EH)
~ people
~ |

P3EH Rate=
if then else(GapP3EH>0,Min(GapP3EH, EH Control)/TimeToMoveIn,GapP3EH/TimeToMoveOut)
~ people/Month
~ |

P3ER= INTEG (
P3ER Rate+P3IRtoER Rate-Attrition Rate P3ER,
StartP3ER)

~ people
~ |

P3ER Rate=
if then else(GapP3ER>0,Min(GapP3ER, ER Control)/TimeToMoveIn,GapP3ER/TimeToMoveOut)
~ people/Month
~ |

P3ET Rate=
if then else(GapP3ET>0,Min(GapP3ET, ET Control)/TimeToMoveIn,GapP3ET/TimeToMoveOut)
~ people/Month
~ |

P3HDesiredE=
DesiredRealHeads P3H*RatioEH
~ people
~ |

P3HDesiredI=
DesiredRealHeads P3H*RatioIH
~ people
~ |

P3HDesiredN=
DesiredRealHeads P3H*RatioNH
~ people
~ |

P3ID= INTEG (
P3ID Rate+P3NDtoID Rate-P3IDtoED Rate-Attrition Rate P3ID,
StartP3ID)
~ people
~ |

P3ID Rate=
if then else(GapP3ID>0,Min(GapP3ID, ID Control)/TimeToMoveIn,GapP3ID/TimeToMoveOut)
~ people/Month
~ |

P3IDtoED Rate=
(P3ID/Intermediate Advance to Expert Time D)*Staffing Gap effect on learning P3D*Complexity effect on learning P3
~ people/Month
~ |

P3IH= INTEG (
P3IH Rate+P3NHtoIH Rate-P3IHtoEH Rate-Attrition Rate P3IH,
StartP3IH)
~ people
~ |

P3IH Rate=
if then else(GapP3IH>0,Min(GapP3IH, IH Control)/TimeToMoveIn,GapP3IH/TimeToMoveOut)
~ people/Month
~ |

P3IHtoEH Rate=
(P3IH/Intermediate Advance to Expert Time H)*Staffing Gap effect on learning P3H*Complexity effect on learning P3
~ people/Month
~ |

P3IR= INTEG (
P3IR Rate+P3NRtoIR Rate-P3IRtoER Rate-Attrition Rate P3IR,
StartP3IR)
~ people
~ |

P3IR Rate=
if then else(GapP3IR>0,Min(GapP3IR, IR Control)/TimeToMoveIn,GapP3IR/TimeToMoveOut)
~ people/Month
~ |

P3IRtoER Rate=
(P3IR/Intermediate Advance to Expert Time R)*Staffing Gap effect on learning P3R

~ people/Month
~ |

P3IT Rate=
if then else(GapP3IT>0,Min(GapP3IT, IT Control)/TimeToMoveIn,GapP3IT/TimeToMoveOut)
~ people/Month
~ |

P3ITtoET Rate=
(P3IT/Intermediate Advance to Expert Time T)*Staffing Gap effect on learning P3T*Complexity effect on learning P3
~ people/Month
~ |

P3ND= INTEG (
P3ND Rate-P3NDtoID Rate-Attrition Rate P3ND,
StartP3ND)
~ people
~ |

P3ND Rate=
if then else(GapP3ND>0,Min(GapP3ND, ND Control)/TimeToMoveIn,GapP3ND/TimeToMoveOut)
~ people/Month
~ |

P3NDtoID Rate=
(P3ND/Novice Advance to Intermediate Time D)*Staffing Gap effect on learning P3D*Complexity effect on learning P3
~ people/Month
~ |

P3NH= INTEG (
P3NH Rate-P3NHtoIH Rate-Attrition Rate P3NH,
StartP3NH)
~ people
~ |

P3NH Rate=
if then else(GapP3NH>0,Min(GapP3NH, NH Control)/TimeToMoveIn,GapP3NH/TimeToMoveOut)
~ people/Month
~ |

P3NHtoIH Rate=
(P3NH/Novice Advance to Intermediate Time H)*Staffing Gap effect on learning P3H*Complexity effect on learning P3
~ people/Month
~ |

P3NR= INTEG (
P3NR Rate-P3NRtoIR Rate-Attrition Rate P3NR,
StartP3NR)
~ people
~ |

P3NR Rate=
if then else(GapP3NR>0,Min(GapP3NR, NR Control)/TimeToMoveIn,GapP3NR/TimeToMoveOut)
~ people/Month
~ |

P3NRtoIR Rate=
(P3NR/Novice Advance to Intermediate Time R)*Staffing Gap effect on learning P3R*Complexity effect on learning P3
~ people/Month
~ |

P3NT Rate=
if then else(GapP3NT>0,Min(GapP3NT, NT Control)/TimeToMoveIn,GapP3NT/TimeToMoveOut)
~ people/Month
~ |

P3NTtoIT Rate=
(P3NT/Novice Advance to Intermediate Time T)*Staffing Gap effect on learning P3T*Complexity effect on learning P3
~ people/Month
~ |

P3RDesiredE=
DesiredRealHeads P3R*RatioER

~ people
~ |

P3RDesiredI=
DesiredRealHeads P3R*RatioIR
~ people
~ |

P3RDesiredN=
DesiredRealHeads P3R*RatioNR
~ people
~ |

P3TDesiredE=
DesiredRealHeads P3T*RatioET
~ people
~ |

P3TDesiredI=
DesiredRealHeads P3T*RatioIT
~ people
~ |

P3TDesiredN=
DesiredRealHeads P3T*RatioNT
~ people
~ |

P4 Initial Priority=
Get XLS Constants('ModelConstants.xls','Project Constants','b5')
~ dmnl
~ |

P4DDesiredE=
DesiredRealHeads P4D*RatioED
~ people
~ |

P4DDesiredI=
DesiredRealHeads P4D*RatioID
~ people
~ |

P4DDesiredN=
DesiredRealHeads P4D*RatioND
~ people
~ |

P4ED= INTEG (
P4ED Rate+P4IDtoED Rate-Attrition Rate P4ED,
StartP4ED)
~ people
~ |

P4ED Rate=
if then else(GapP4ED>0,Min(GapP4ED, ED Control)/TimeToMoveIn,GapP4ED/TimeToMoveOut)
~ people/Month
~ |

P4EH= INTEG (
P4EH Rate+P4IHtoEH Rate-Attrition Rate P4EH,
StartP4EH)
~ people
~ |

P4EH Rate=
if then else(GapP4EH>0,Min(GapP4EH, EH Control)/TimeToMoveIn,GapP4EH/TimeToMoveOut)
~ people/Month
~ |

P4ER= INTEG (
P4ER Rate+P4IRtoER Rate-Attrition Rate P4ER,
StartP4ER)

~ people
~ |

P4ER Rate=
if then else(GapP4ER>0,Min(GapP4ER, ER Control)/TimeToMoveIn,GapP4ER/TimeToMoveOut)
~ people/Month
~ |

P4ET Rate=
if then else(GapP4ET>0,Min(GapP4ET, ET Control)/TimeToMoveIn,GapP4ET/TimeToMoveOut)
~ people/Month
~ |

P4HDesiredE=
DesiredRealHeads P4H*RatioEH
~ people
~ |

P4HDesiredI=
DesiredRealHeads P4H*RatioIH
~ people
~ |

P4HDesiredN=
DesiredRealHeads P4H*RatioNH
~ people
~ |

P4ID= INTEG (
P4ID Rate+P4NDtoID Rate-P4IDtoED Rate-Attrition Rate P4ID,
StartP4ID)
~ people
~ |

P4ID Rate=
if then else(GapP4ID>0,Min(GapP4ID, ID Control)/TimeToMoveIn,GapP4ID/TimeToMoveOut)
~ people/Month
~ |

P4IDtoED Rate=
(P4ID/Intermediate Advance to Expert Time D)*Staffing Gap effect on learning P4D*Complexity effect on learning P4
~ people/Month
~ |

P4IH= INTEG (
P4IH Rate+P4NHtoIH Rate-P4IHtoEH Rate-Attrition Rate P4IH,
StartP4IH)
~ people
~ |

P4IH Rate=
if then else(GapP4IH>0,Min(GapP4IH,IH Control)/TimeToMoveIn,GapP4IH/TimeToMoveOut)
~ people/Month
~ |

P4IHtoEH Rate=
(P4IH/Intermediate Advance to Expert Time H)*Staffing Gap effect on learning P4H*Complexity effect on learning P4
~ people/Month
~ |

P4IR= INTEG (
P4IR Rate+P4NRtoIR Rate-P4IRtoER Rate-Attrition Rate P4IR,
StartP4IR)
~ people
~ |

P4IR Rate=
if then else(GapP4IR>0,Min(GapP4IR, IR Control)/TimeToMoveIn,GapP4IR/TimeToMoveOut)
~ people/Month
~ |

P4IRtoER Rate=
(P4IR/Intermediate Advance to Expert Time R)*Staffing Gap effect on learning P4R*Complexity effect on learning P4

~ people/Month
~ |

P4IT Rate=
if then else(GapP4IT>0,Min(GapP4IT, IT Control)/TimeToMoveIn,GapP4IT/TimeToMoveOut)
~ people/Month
~ |

P4ITtoET Rate=
(P4IT/Intermediate Advance to Expert Time T)*Staffing Gap effect on learning P4T*Complexity effect on learning P4
~ people/Month
~ |

P4ND= INTEG (
P4ND Rate-P4NDtoID Rate-Attrition Rate P4ND,
StartP4ND)
~ people
~ |

P4ND Rate=
if then else(GapP4ND>0,Min(GapP4ND, ND Control)/TimeToMoveIn,GapP4ND/TimeToMoveOut)
~ people/Month
~ |

P4NDtoID Rate=
(P4ND/Novice Advance to Intermediate Time D)*Staffing Gap effect on learning P4D*Complexity effect on learning P4
~ people/Month
~ |

P4NH= INTEG (
P4NH Rate-P4NHtoIH Rate-Attrition Rate P4NH,
StartP4NH)
~ people
~ |

P4NH Rate=
if then else(GapP4NH>0,Min(GapP4NH, NH Control)/TimeToMoveIn,GapP4NH/TimeToMoveOut)
~ people/Month
~ |

P4NHtoIH Rate=
(P4NH/Novice Advance to Intermediate Time H)*Staffing Gap effect on learning P4H*Complexity effect on learning P4
~ people/Month
~ |

P4NR= INTEG (
P4NR Rate-P4NRtoIR Rate-Attrition Rate P4NR,
StartP4NR)
~ people
~ |

P4NR Rate=
if then else(GapP4NR>0,Min(GapP4NR, NR Control)/TimeToMoveIn,GapP4NR/TimeToMoveOut)
~ people/Month
~ |

P4NRtoIR Rate=
(P4NR/Novice Advance to Intermediate Time R)*Staffing Gap effect on learning P4R*Complexity effect on learning P4
~ people/Month
~ |

P4NT Rate=
if then else(GapP4NT>0,Min(GapP4NT, NT Control)/TimeToMoveIn,GapP4NT/TimeToMoveOut)
~ people/Month
~ |

P4NTtoIT Rate=
(P4NT/Novice Advance to Intermediate Time T)*Staffing Gap effect on learning P4T*Complexity effect on learning P4
~ people/Month
~ |

P4RDesiredE=
DesiredRealHeads P4R*RatioER

~ people
~ |

P4RDesiredI=
DesiredRealHeads P4R*RatioIR
~ people
~ |

P4RDesiredN=
DesiredRealHeads P4R*RatioNR
~ people
~ |

P4TDesiredE=
DesiredRealHeads P4T*RatioET
~ people
~ |

P4TDesiredI=
DesiredRealHeads P4T*RatioIT
~ people
~ |

P4TDesiredN=
DesiredRealHeads P4T*RatioNT
~ people
~ |

PDY P1D=
(Normal Productivity D*Fatigue effect PDY P1D)*Complexity effect on PDY P1
~ lines/(people*Month)
~ |

PDY P1T=
Normal Productivity T*Fatigue effect PDY P1T*Complexity effect on PDY P1
~ lines/people/Month
~ |

PDY P2D=
Normal Productivity D*Fatigue effect PDY P2D*Complexity effect on PDY P2
~ lines/(people*Month)
~ |

PDY P2H=
Normal Productivity H*Fatigue effect PDY P2H*Complexity effect on PDY P2
~ lines/(people*Month)
~ |

PDY P2R=
Normal Productivity R*Fatigue effect PDY P2R*Complexity effect on PDY P2
~ lines/(people*Month)
~ |

PDY P2T=
Normal Productivity T*Fatigue effect PDY P2T*Complexity effect on PDY P2
~ lines/(people*Month)
~ |

PDY P3D=
Normal Productivity D*Fatigue effect PDY P3D*Complexity effect on PDY P3
~ lines/people/Month
~ |

PDY P3H=
Normal Productivity H*Fatigue effect PDY P3H*Complexity effect on PDY P3
~ lines/(people*Month)
~ |

PDY P3R=
Normal Productivity R*Fatigue effect PDY P3R*Complexity effect on PDY P3
~ lines/(people*Month)
~ |

PDY P3T=
Normal Productivity T*Fatigue effect PDY P3T*Complexity effect on PDY P3
~ lines/people/Month
~ |

PDY P4D=
Normal Productivity D*Fatigue effect PDY P4D*Complexity effect on PDY P4
~ lines/(people*Month)
~ |

PDY P4H=
Normal Productivity H*Fatigue effect PDY P4H*Complexity effect on PDY P4
~ lines/(people*Month)
~ |

PDY P4R=
Normal Productivity R*Fatigue effect PDY P4R*Complexity effect on PDY P4
~ lines/people/Month
~ |

PDY P4T=
Normal Productivity T*Fatigue effect PDY P4T*Complexity effect on PDY P4
~ lines/(people*Month)
~ |

Percvd PDY P1D =INTEG((PDY P1D - Percvd PDY P1D)/TimeToPercvPDY D,Normal Productivity D\
)
~ lines/(people*Month)
~ |

Percvd PDY P1H =INTEG((PDY P1H - Percvd PDY P1H)/TimeToPercvPDY H,Normal Productivity H\
)
~ lines/(people*Month)
~ |

Percvd PDY P1R =INTEG((PDY P1R - Percvd PDY P1R)/TimeToPercvPDY R,Normal Productivity R\
)
~ lines/(people*Month)
~ |

Percvd PDY P1T =INTEG((PDY P1T - Percvd PDY P1T)/TimeToPercvPDY T,Normal Productivity T\
)
~ lines/(people*Month)
~ |

Priority effect on attractiveness f(
[(0,0)-(10,10)],(0,0),(10,10))
~ dmnl
~ |

Priority effect on attractiveness P1D=
Priority effect on attractiveness f(P1 Initial Priority)
~ dmnl
~ |

Priority effect on attractiveness P1H=
Priority effect on attractiveness f(P1 Initial Priority)
~ dmnl
~ |

Priority effect on attractiveness P1R=
Priority effect on attractiveness f(P1 Initial Priority)
~ dmnl
~ |

Priority effect on attractiveness P1T=
Priority effect on attractiveness f(P1 Initial Priority)
~ dmnl
~ |

Priority effect on attractiveness P2D=
Priority effect on attractiveness f(P2 Initial Priority)
~ dmnl

~ |

Priority effect on attractiveness P2H=
Priority effect on attractiveness f(P2 Initial Priority)
~ dmnl
~ |

Priority effect on attractiveness P2R=
Priority effect on attractiveness f(P2 Initial Priority)
~ dmnl
~ |

Priority effect on attractiveness P2T=
Priority effect on attractiveness f(P2 Initial Priority)
~ dmnl
~ |

Priority effect on attractiveness P3D=
Priority effect on attractiveness f(P3 Initial Priority)
~ dmnl
~ |

Priority effect on attractiveness P3H=
Priority effect on attractiveness f(P3 Initial Priority)
~ dmnl
~ |

Priority effect on attractiveness P3R=
Priority effect on attractiveness f(P3 Initial Priority)
~ dmnl
~ |

Priority effect on attractiveness P3T=
Priority effect on attractiveness f(P3 Initial Priority)
~ dmnl
~ |

Priority effect on attractiveness P4D=
Priority effect on attractiveness f(P4 Initial Priority)
~ dmnl
~ |

Priority effect on attractiveness P4H=
Priority effect on attractiveness f(P4 Initial Priority)
~ dmnl
~ |

Priority effect on attractiveness P4R=
Priority effect on attractiveness f(P4 Initial Priority)
~ dmnl
~ |

Priority effect on attractiveness P4T=
Priority effect on attractiveness f(P4 Initial Priority)
~ dmnl
~ |

Priority Weight=
Get XLS Constants('ModelConstants.xls','Portfolio Constants','b2')
~ dmnl
~ |

project:
one,two,three,four
~
~ |

Qual P1D=
Min(1, MaxQuality D*Fatigue effect qual P1D*Average Skill Effect on Quality P1D*Complexity effect on quality P1\
)
~ fraction
~ |

Qual P1H=
Min(1, MaxQuality H*Fatigue effect qual P1H*Average Skill Effect on Quality P1H*Complexity effect on quality P1\
)
~ fraction
~ |

Qual P1T=
Min(1, MaxQuality T*Fatigue effect qual P1T*Average Skill Effect on Quality P1T*Complexity effect on quality P1\
)
~ fraction
~ |

Remaining Time P1D=
DueDate P1D - Time
~ Month
~ |

Remaining Time P1H=
DueDate P1H - Time
~ Month
~ |

Remaining Time P1R=
DueDate P1R - Time
~ Month
~ |

Remaining Time P1T=
DueDate P1T - Time
~ Month
~ |

Remaining Time P2D=
DueDate P2D - Time
~ Month
~ |

Remaining Time P2H=
DueDate P2H - Time
~ Month
~ |

Remaining Time P2R=
DueDate P2R - Time
~ Month
~ |

Remaining Time P2T=
DueDate P2T - Time
~ Month
~ |

Remaining Time P3D=
DueDate P3D - Time
~ Month
~ |

Remaining Time P3H=
DueDate P3H - Time
~ Month
~ |

Remaining Time P3R=
DueDate P3R - Time
~ Month
~ |

Remaining Time P3T=
DueDate P3T - Time
~ Month
~ |

Remaining Time P4D=

```
	DueDate P4D - Time
	~	Month
	~		|

Remaining Time P4H=
	DueDate P4H - Time
	~	Month
	~		|

Remaining Time P4R=
	DueDate P4R - Time
	~	Month
	~		|

Remaining Time P4T=
	DueDate P4T - Time
	~	Month
	~		|

Staffing Gap effect on attractiveness f(
	[(0,0)-(1e+009,1)],(0,1),(0.5,0.9),(1,0.8),(2,0.5),(4,0.1),(10,0),(1e+009,0))
	~	dmnl
	~		|

Staffing Gap effect on attractiveness P1D=
	Staffing Gap effect on attractiveness f(ZIDZ(Workforce P1D,DesiredPeople P1D))
	~	dmnl
	~		|

Staffing Gap effect on attractiveness P1H=
	Staffing Gap effect on attractiveness f(ZIDZ(Workforce P1H,DesiredPeople P1H))
	~	dmnl
	~		|

Staffing Gap effect on attractiveness P1R=
	Staffing Gap effect on attractiveness f(ZIDZ(Workforce P1R,DesiredPeople P1R))
	~	dmnl
	~		|

Staffing Gap effect on attractiveness P1T=
	Staffing Gap effect on attractiveness f(ZIDZ(Workforce P1T,DesiredPeople P1T))
	~	dmnl
	~		|

Staffing Gap effect on attractiveness P2D=
	Staffing Gap effect on attractiveness f(ZIDZ(Workforce P2D,DesiredPeople P2D))
	~	dmnl
	~		|

Staffing Gap effect on attractiveness P2H=
	Staffing Gap effect on attractiveness f(ZIDZ(Workforce P2H,DesiredPeople P2H))
	~	dmnl
	~		|

Staffing Gap effect on attractiveness P2R=
	Staffing Gap effect on attractiveness f(ZIDZ(Workforce P2R,DesiredPeople P2R))
	~	dmnl
	~		|

Staffing Gap effect on attractiveness P2T=
	Staffing Gap effect on attractiveness f(ZIDZ(Workforce P2T,DesiredPeople P2T))
	~	dmnl
	~		|

Staffing Gap effect on attractiveness P3D=
	Staffing Gap effect on attractiveness f(ZIDZ(Workforce P3D,DesiredPeople P3D))
	~	dmnl
	~		|

Staffing Gap effect on attractiveness P3H=
	Staffing Gap effect on attractiveness f(ZIDZ(Workforce P3H,DesiredPeople P3H))
	~	dmnl
	~		|
```

Staffing Gap effect on attractiveness P3R=
Staffing Gap effect on attractiveness f(ZIDZ(Workforce P3R,DesiredPeople P3R))
~ dmnl
~ |

Staffing Gap effect on attractiveness P3T=
Staffing Gap effect on attractiveness f(ZIDZ(Workforce P3T,DesiredPeople P3T))
~ dmnl
~ |

Staffing Gap effect on attractiveness P4D=
Staffing Gap effect on attractiveness f(ZIDZ(Workforce P4D,DesiredPeople P4D))
~ dmnl
~ |

Staffing Gap effect on attractiveness P4H=
Staffing Gap effect on attractiveness f(ZIDZ(Workforce P4H,DesiredPeople P4H))
~ dmnl
~ |

Staffing Gap effect on attractiveness P4R=
Staffing Gap effect on attractiveness f(ZIDZ(Workforce P4R,DesiredPeople P4R))
~ dmnl
~ |

Staffing Gap effect on attractiveness P4T=
Staffing Gap effect on attractiveness f(ZIDZ(Workforce P4T,DesiredPeople P4T))
~ dmnl
~ |

Staffing Gap Weight=
Get XLS Constants('ModelConstants.xls','Portfolio Constants','b4')
~ dmnl
~ |

Start Dev Rate P1D=
if then else(Initial WorkToDo P1D>0,(Doing right P1H/Init HLD P1H)*Init Dev P1D,0)
~ lines/Month
~ |

Start Dev Rate P2D=
if then else(Initial WorkToDo P2D>0,(Doing right P2H/Init HLD P2H)*Init Dev P2D,0)
~ lines/Month
~ |

Start Dev Rate P3D=
if then else(Initial WorkToDo P3D>0,(Doing right P3H/Init HLD P3H)*Init Dev P3D,0)
~ lines/Month
~ |

Start Dev Rate P4D=
if then else(Initial WorkToDo P4D>0,(Doing right P4H/Init HLD P4H)*Init Dev P4D,0)
~ lines/Month
~ |

StartP1ED=
Get XLS Constants('ModelConstants.xls','Project Constants','k2')
~ people
~ |

StartP1EH=
Get XLS Constants('ModelConstants.xls','Project Constants','h2')
~ people
~ |

StartP1ER=
Get XLS Constants('ModelConstants.xls','Project Constants','e2')
~ people
~ |

StartP1ET=
Get XLS Constants('ModelConstants.xls','Project Constants','n2')

~ people
~ |

StartP1ID=
Get XLS Constants('ModelConstants.xls','Project Constants','j2')
~ people
~ |

StartP1IH=
Get XLS Constants('ModelConstants.xls','Project Constants','g2')
~ people
~ |

StartP1IR=
Get XLS Constants('ModelConstants.xls','Project Constants','d2')
~ people
~ |

StartP1IT=
Get XLS Constants('ModelConstants.xls','Project Constants','m2')
~ people
~ |

StartP1ND=
Get XLS Constants('ModelConstants.xls','Project Constants','i2')
~ people
~ |

StartP1NH=
Get XLS Constants('ModelConstants.xls','Project Constants','f2')
~ people
~ |

StartP1NR=
Get XLS Constants('ModelConstants.xls','Project Constants','c2')
~ people
~ |

StartP1NT=
Get XLS Constants('ModelConstants.xls','Project Constants','l2')
~ people
~ |

StartP2ED=
Get XLS Constants('ModelConstants.xls','Project Constants','k3')
~ people
~ |

StartP2EH=
Get XLS Constants('ModelConstants.xls','Project Constants','h3')
~ people
~ |

StartP2ER=
Get XLS Constants('ModelConstants.xls','Project Constants','e3')
~ people
~ |

StartP2ET=
Get XLS Constants('ModelConstants.xls','Project Constants','n3')
~ people
~ |

StartP2ID=
Get XLS Constants('ModelConstants.xls','Project Constants','j3')
~ people
~ |

StartP2IH=
Get XLS Constants('ModelConstants.xls','Project Constants','g3')
~ people
~ |

```
StartP2IR=
        Get XLS Constants('ModelConstants.xls','Project Constants','d3')
        ~       people
        ~               |

StartP2IT=
        Get XLS Constants('ModelConstants.xls','Project Constants','m3')
        ~       people
        ~               |

StartP2ND=
        Get XLS Constants('ModelConstants.xls','Project Constants','i3')
        ~       people
        ~               |

StartP2NH=
        Get XLS Constants('ModelConstants.xls','Project Constants','f3')
        ~       people
        ~               |

StartP2NR=
        Get XLS Constants('ModelConstants.xls','Project Constants','c3')
        ~       people
        ~               |

StartP2NT=
        Get XLS Constants('ModelConstants.xls','Project Constants','l3')
        ~       people
        ~               |

StartP3ED=
        Get XLS Constants('ModelConstants.xls','Project Constants','k4')
        ~       people
        ~               |

StartP3EH=
        Get XLS Constants('ModelConstants.xls','Project Constants','h4')
        ~       people
        ~               |

StartP3ER=
        Get XLS Constants('ModelConstants.xls','Project Constants','e4')
        ~       people
        ~               |

StartP3ET=
        Get XLS Constants('ModelConstants.xls','Project Constants','n4')
        ~       people
        ~               |

StartP3ID=
        Get XLS Constants('ModelConstants.xls','Project Constants','j4')
        ~       people
        ~               |

StartP3IH=
        Get XLS Constants('ModelConstants.xls','Project Constants','g4')
        ~       people
        ~               |

StartP3IR=
        Get XLS Constants('ModelConstants.xls','Project Constants','d4')
        ~       people
        ~               |

StartP3IT=
        Get XLS Constants('ModelConstants.xls','Project Constants','m4')
        ~       people
        ~               |

StartP3ND=
        Get XLS Constants('ModelConstants.xls','Project Constants','i4')
        ~       people
```

```
~ |

StartP3NH=
	Get XLS Constants('ModelConstants.xls','Project Constants','f4')
	~	people
	~		|

StartP3NR=
	Get XLS Constants('ModelConstants.xls','Project Constants','c4')
	~	people
	~		|

StartP3NT=
	Get XLS Constants('ModelConstants.xls','Project Constants','l4')
	~	people
	~		|

StartP4ED=
	Get XLS Constants('ModelConstants.xls','Project Constants','k5')
	~	people
	~		|

StartP4EH=
	Get XLS Constants('ModelConstants.xls','Project Constants','h5')
	~	people
	~		|

StartP4ER=
	Get XLS Constants('ModelConstants.xls','Project Constants','e5')
	~	people
	~		|

StartP4ET=
	Get XLS Constants('ModelConstants.xls','Project Constants','n5')
	~	people
	~		|

StartP4ID=
	Get XLS Constants('ModelConstants.xls','Project Constants','j5')
	~	people
	~		|

StartP4IH=
	Get XLS Constants('ModelConstants.xls','Project Constants','g5')
	~	people
	~		|

StartP4IR=
	Get XLS Constants('ModelConstants.xls','Project Constants','d5')
	~	people
	~		|

StartP4IT=
	Get XLS Constants('ModelConstants.xls','Project Constants','m5')
	~	people
	~		|

StartP4ND=
	Get XLS Constants('ModelConstants.xls','Project Constants','i5')
	~	people
	~		|

StartP4NH=
	Get XLS Constants('ModelConstants.xls','Project Constants','f5')
	~	people
	~		|

StartP4NR=
	Get XLS Constants('ModelConstants.xls','Project Constants','c5')
	~	people
	~		|

StartP4NT=
```

```
        Get XLS Constants('ModelConstants.xls','Project Constants','l5')
        ~       people
        ~               |

Test Start Rate P1T=
        if then else(Initial WorkToDo P1T>0,(Doing right P1D/Init Dev P1D)*Init Test P1T,0)
        ~       lines/Month
        ~               |

Test Start Rate P2T=
        if then else(Initial WorkToDo P2T>0,(Doing right P2D/Init Dev P2D)*Init Test P2T,0)
        ~       lines/Month
        ~               |

Test Start Rate P3T=
        if then else(Initial WorkToDo P3T>0,(Doing right P3D/Init Dev P3D)*Init Test P3T,0)
        ~       lines/Month
        ~               |

Test Start Rate P4T=
        if then else(Initial WorkToDo P4T>0,(Doing right P4D/Init Dev P4D)*Init Test P4T,0)
        ~       lines/Month
        ~               |

Time for Attrition=
        Get XLS Constants('ModelConstants.xls','Portfolio Constants','b13')
        ~       Month
        ~               |

Time to retire=
        1
        ~       Month
        ~               |

TimeToGetFatigued D=
        Get XLS Constants('ModelConstants.xls','Phase Constants','d5')
        ~       Month
        ~       Should be 0.25 or so?
        |

TimeToGetFatigued H=
        Get XLS Constants('ModelConstants.xls','Phase Constants','c5')
        ~       Month
        ~       Should be 0.25 or so?
        |

TimeToGetFatigued R=
        Get XLS Constants('ModelConstants.xls','Phase Constants','b5')
        ~       Month
        ~       Should be 0.25 or so?
        |

TimeToGetFatigued T=
        Get XLS Constants('ModelConstants.xls','Phase Constants','e5')
        ~       Month
        ~       Should be 0.25 or so?
        |

TimeToMoveIn=
        Get XLS Constants('ModelConstants.xls','Portfolio Constants','b9')
        ~       Month
        ~               |

TimeToMoveOut=
        Get XLS Constants('ModelConstants.xls','Portfolio Constants','b10')
        ~       Month
        ~               |

TimeToPercvPDY D=
        Get XLS Constants('ModelConstants.xls','Phase Constants','d6')
        ~       Month
        ~               |
```

```
TimeToPercvPDY H=
	Get XLS Constants('ModelConstants.xls','Phase Constants','c6')
	~	Month
	~		|

TimeToPercvPDY R=
	Get XLS Constants('ModelConstants.xls','Phase Constants','b6')
	~	Month
	~		|

TimeToPercvPDY T=
	Get XLS Constants('ModelConstants.xls','Phase Constants','e6')
	~	Month
	~		|

TimeToStart P1R=
	Get XLS Constants('ModelConstants.xls','Project Constants','w2')
	~	Month
	~		|

TimeToStart P2R=
	Get XLS Constants('ModelConstants.xls','Project Constants','w3')
	~	Month
	~		|

TimeToStart P3R=
	Get XLS Constants('ModelConstants.xls','Project Constants','w4')
	~	Month
	~		|

TimeToStart P4R=
	Get XLS Constants('ModelConstants.xls','Project Constants','w5')
	~	Month
	~		|

Total Attrition ED= INTEG (
	Attrition Rate P1ED+Attrition Rate P2ED+Attrition Rate P3ED+Attrition Rate P4ED,
		0)
	~	people
	~		|

Total Attrition EH= INTEG (
	Attrition Rate P1EH+Attrition Rate P2EH+Attrition Rate P3EH+Attrition Rate P4EH,
		0)
	~	people
	~		|

Total Attrition ID= INTEG (
	Attrition Rate P1ID+Attrition Rate P2ID+Attrition Rate P3ID+Attrition Rate P4ID,
		0)
	~	people
	~		|

Total Attrition IH= INTEG (
	Attrition Rate P1IH+Attrition Rate P2IH+Attrition Rate P3IH+Attrition Rate P4IH,
		0)
	~	people
	~		|

Total Attrition IR= INTEG (
	Attrition Rate P1IR+Attrition Rate P2IR+Attrition Rate P3IR+Attrition Rate P4IR,
		0)
	~	people
	~		|

Total Attrition NH= INTEG (
	Attrition Rate P1NH+Attrition Rate P2NH+Attrition Rate P3NH+Attrition Rate P4NH,
		0)
	~	people
	~		|

Total Attrition NR= INTEG (
	Attrition Rate P1NR+Attrition Rate P2NR+Attrition Rate P3NR+Attrition Rate P4NR,
```

0)
~ people
~ |

Total Attrition ER= INTEG (
Attrition Rate P1ER+Attrition Rate P2ER+Attrition Rate P3ER+Attrition Rate P4ER,
0)
~ people
~ |

Total Attritition ND= INTEG (
Attrition Rate P1ND+Attrition Rate P2ND+Attrition Rate P3ND+Attrition Rate P4ND,
0)
~ people
~ |

TotalDesiredP1=
DesiredP1NR+DesiredP1IR+DesiredP1ER+DesiredP1NH+DesiredP1IH+DesiredP1EH+DesiredP1ND+\
DesiredP1ID+DesiredP1ED+DesiredP1NT+DesiredP1IT+DesiredP1ET
~ people
~ |

TotalDesiredP2=
DesiredP2NR+DesiredP2IR+DesiredP2ER+DesiredP2NH+DesiredP2IH+DesiredP2EH+DesiredP2ND+\
DesiredP2ID+DesiredP2ED+DesiredP2NT+DesiredP2IT+DesiredP2ET
~ people
~ |

TotalDesiredP3=
DesiredP3NR+DesiredP3IR+DesiredP3ER+DesiredP3NH+DesiredP3IH+DesiredP3EH+DesiredP3ND+\
DesiredP3ID+DesiredP3ED+DesiredP3NT+DesiredP3IT+DesiredP3ET
~ people
~ |

TotalDesiredP4=
DesiredP4NR+DesiredP4IR+DesiredP4ER+DesiredP4NH+DesiredP4IH+DesiredP4EH+DesiredP4ND+\
DesiredP4ID+DesiredP4ED+DesiredP4NT+DesiredP4IT+DesiredP4ET
~ people
~ |

TotalED=
(P1ED+P2ED+ED Control+P3ED+P4ED)
~ people
~ |

TotalEH=
(P1EH+P2EH+EH Control+P3EH+P4EH)
~ people
~ |

TotalER=
(P1ER+P2ER+ER Control+P3ER+P4ER)
~ people
~ |

TotalET=
(P1ET+P2ET+ET Control+P3ET+P4ET)
~ people
~ |

TotalGap D=
SUM(NDDesired[project!])-TotalND+SUM(IDDesired[project!])-TotalID+SUM(EDDesired[project\
!])-TotalED
~ people
~ |

TotalGap H=
SUM(NHDesired[project!])-TotalNH+SUM(IHDesired[project!])-TotalIH+SUM(EHDesired[project\
!])-TotalEH
~ people
~ |

TotalGap R=

```
	SUM(NRDesired[project!])-TotalNR+SUM(IRDesired[project!])-TotalIR+SUM(ERDesired[project\
		!])-TotalER
	~	people
	~		|

TotalID=
	P1ID+P2ID+ID Control+P3ID+P4ID
	~	people
	~		|

TotalIH=
	P1IH+P2IH+IH Control+P3IH+P4IH
	~	people
	~		|

TotalIT=
	P1IT+P2IT+IT Control+P3IT+P4IT
	~	people
	~		|

TotalND=
	P1ND+P2ND+ND Control+P3ND+P4ND
	~	people
	~		|

TotalNH=
	P1NH+P2NH+NH Control+P3NH+P4NH
	~	people
	~		|

TotalNR=
	P1NR+P2NR+NR Control+P3NR+P4NR
	~	people
	~		|

TotalNT=
	P1NT+P2NT+NT Control+P3NT+P4NT
	~	people
	~		|

TotalP1=
	P1NR+P1IR+P1ER+P1NH+P1IH+P1EH+P1ND+P1ID+P1ED+P1NT+P1IT+P1ET
	~	people
	~		|

TotalP3=
	P3NR+P3IR+P3ER+P3NH+P3IH+P3EH+P3ND+P3ID+P3ED+P3NT+P3IT+P3ET
	~	people
	~		|

TotalP4=
	P4NR+P4IR+P4ER+P4NH+P4IH+P4EH+P4ND+P4ID+P4ED+P4NT+P4IT+P4ET
	~	people
	~		|

Workforce P1D=
	if then else(P1ND*NoviceMultiplier P1D+P1ID*IntMultiplier P1D+P1ED*ExpertMultiplier P1D\
		<0.01,0.01,P1ND*NoviceMultiplier P1D+P1ID*IntMultiplier P1D+P1ED*ExpertMultiplier P1D\
		)
	~	people
	~		|

Workforce P1H=
	if then else(P1NH*NoviceMultiplier P1H+P1IH*IntMultiplier P1H+P1EH*ExpertMultiplier P1H\
		<0.01,0.01,P1NH*NoviceMultiplier P1H+P1IH*IntMultiplier P1H+P1EH*ExpertMultiplier P1H\
		)
	~	people
	~		|

Workforce P1R=
	if then else(P1NR*NoviceMultiplier P1R+P1IR*IntMultiplier P1R+P1ER*ExpertMultiplier P1R\
		<0.01,0.01,P1NR*NoviceMultiplier P1R+P1IR*IntMultiplier P1R+P1ER*ExpertMultiplier P1R\
		)
```

```
	~	people
	~		|

Workforce P1T=
	if then else(P1NT*NoviceMultiplier P1T+P1IT*IntMultiplier P1T+P1ET*ExpertMultiplier P1T\
		<0.01,0.01,P1NT*NoviceMultiplier P1T+P1IT*IntMultiplier P1T+P1ET*ExpertMultiplier P1T\
		)
	~	people
	~		|

Workforce P2D=
	if then else(P2ND*NoviceMultiplier P2D+P2ID*IntMultiplier P2D+P2ED*ExpertMultiplier P2D\
		<0.01,0.01,P2ND*NoviceMultiplier P2D+P2ID*IntMultiplier P2D+P2ED*ExpertMultiplier P2D\
		)
	~	people
	~		|

Workforce P2H=
	if then else(P2NH*NoviceMultiplier P2H+P2IH*IntMultiplier P2H+P2EH*ExpertMultiplier P2H\
		<0.01,0.01,P2NH*NoviceMultiplier P2H+P2IH*IntMultiplier P2H+P2EH*ExpertMultiplier P2H\
		)
	~	people
	~		|

Workforce P2R=
	if then else(P2NR*NoviceMultiplier P2R+P2IR*IntMultiplier P2R+P2ER*ExpertMultiplier P2R\
		<0.01,0.01,P2NR*NoviceMultiplier P2R+P2IR*IntMultiplier P2R+P2ER*ExpertMultiplier P2R\
		)
	~	people
	~		|

Workforce P2T=
	if then else(P2NT*NoviceMultiplier P2T+P2IT*IntMultiplier P2T+P2ET*ExpertMultiplier P2T\
		<0.01,0.01,P2NT*NoviceMultiplier P2T+P2IT*IntMultiplier P2T+P2ET*ExpertMultiplier P2T\
		)
	~	people
	~		|

Workforce P3D=
	if then else(P3ND*NoviceMultiplier P3D+P3ID*IntMultiplier P3D+P3ED*ExpertMultiplier P3D\
		<0.01,0.01,P3ND*NoviceMultiplier P3D+P3ID*IntMultiplier P3D+P3ED*ExpertMultiplier P3D\
		)
	~	people
	~		|

Workforce P3H=
	if then else(P3NH*NoviceMultiplier P3H+P3IH*IntMultiplier P3H+P3EH*ExpertMultiplier P3H\
		<0.01,0.01,P3NH*NoviceMultiplier P3H+P3IH*IntMultiplier P3H+P3EH*ExpertMultiplier P3H\
		)
	~	people
	~		|

Workforce P3R=
	if then else(P3NR*NoviceMultiplier P3R+P3IR*IntMultiplier P3R+P3ER*ExpertMultiplier P3R\
		<0.01,0.01,P3NR*NoviceMultiplier P3R+P3IR*IntMultiplier P3R+P3ER*ExpertMultiplier P3R\
		)
	~	people
	~		|

Workforce P3T=
	if then else(P3NT*NoviceMultiplier P3T+P3IT*IntMultiplier P3T+P3ET*ExpertMultiplier P3T\
		<0.01,0.01,P3NT*NoviceMultiplier P3T+P3IT*IntMultiplier P3T+P3ET*ExpertMultiplier P3T\
		)
	~	people
	~		|

Workforce P4D=
	if then else(P4ND*NoviceMultiplier P4D+P4ID*IntMultiplier P4D+P4ED*ExpertMultiplier P4D\
		<0.01,0.01,P4ND*NoviceMultiplier P4D+P4ID*IntMultiplier P4D+P4ED*ExpertMultiplier P4D\
		)
	~	people
	~		|
```

```
Workforce P4H=
	if then else(P4NH*NoviceMultiplier P4H+P4IH*IntMultiplier P4H+P4EH*ExpertMultiplier P4H\
		<0.01,0.01,P4NH*NoviceMultiplier P4H+P4IH*IntMultiplier P4H+P4EH*ExpertMultiplier P4H\
		)
	~	people
	~		|

Workforce P4R=
	if then else(P4NR*NoviceMultiplier P4R+P4IR*IntMultiplier P4R+P4ER*ExpertMultiplier P4R\
		<0.01,0.01,P4NR*NoviceMultiplier P4R+P4IR*IntMultiplier P4R+P4ER*ExpertMultiplier P4R\
		)
	~	people
	~		|

Workforce P4T=
	if then else(P4NT*NoviceMultiplier P4T+P4IT*IntMultiplier P4T+P4ET*ExpertMultiplier P4T\
		<0.01,0.01,P4NT*NoviceMultiplier P4T+P4IT*IntMultiplier P4T+P4ET*ExpertMultiplier P4T\
		)
	~	people
	~		|

WorkToDo P1D= INTEG (
	FindBugs P1D-Doing P1D+Start Dev Rate P1D,
		0)
	~	lines
	~		|

WorkToDo P1H= INTEG (
	FindBugs P1H-Doing P1H+HLD Start Rate P1H,
		0)
	~	lines
	~		|

WorkToDo P1R= INTEG (
	FindBugs P1R-Doing P1R+InitialWorkToDo P1R Pulse,
		0)
	~	lines
	~		|

WorkToDo P1T= INTEG (
	FindBugs P1T-Doing P1T+Test Start Rate P1T,
		0)
	~	lines
	~		|

WorkToDo P2R= INTEG (
	FindBugs P2R-Doing P2R+InitialWorkToDo P2R Pulse,
		0)
	~	lines
	~		|

WorkToDo P3R= INTEG (
	FindBugs P3R-Doing P3R+InitialWorkToDo P3R Pulse,
		0)
	~	lines
	~		|

WorkToDo P4R= INTEG (
	FindBugs P4R-Doing P4R+InitialWorkToDo P4R Pulse,
		0)
	~	lines
	~		|

********************************************************
	.Control
********************************************************~
		Simulation Control Parameters
	|

AllocatedP4NR=
	NRAllocated[four]
	~	people
	~		|
```


5305-99

FINAL TIME = 60
~ Month
~ The final time for the simulation.
|

GapP2IR=
AllocatedP2IR-P2IR
~ people
~ |

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

P2IDtoED Rate=
(P2ID/Intermediate Advance to Expert Time D)*Staffing Gap effect on learning P2D*Complexity effect on learning P2
~ people/Month
~ |

P2IHtoEH Rate=
(P2IH/Intermediate Advance to Expert Time H)*Staffing Gap effect on learning P2H*Complexity effect on learning P2
~ people/Month
~ |

P2IRtoER Rate=
(P2IR/Intermediate Advance to Expert Time R)*Staffing Gap effect on learning P2R*Complexity effect on learning P2
~ people/Month
~ |

P2ITtoET Rate=
(P2IT/Intermediate Advance to Expert Time T)*Staffing Gap effect on learning P2T*Complexity effect on learning P2
~ people/Month
~ |

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

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

TotalIR=
P1IR+P2IR+IR Control+P3IR+P4IR
~ people
~ |

TotalP2=
P2NR+P2IR+P2ER+P2NH+P2IH+P2EH+P2ND+P2ID+P2ED+P2NT+P2IT+P2ET
~ people
~ |