PRIORITIZATION AND OPTIMIZATION OF SOFTWARE DEPLOYMENT

Rationale For Large Scale Software Upgrade to Enhance Performance and Deployment Methodologies to Minimize Time to Completion, and the Effects of Managerial and Political Interference in Information Technology Strategic Decisions.

By

Dominic Howard

A Research Paper
Submitted in Partial Fulfillment of the
Requirements for the
Master of Science Degree
With a Major in

Management Technology
Information Technology Operational Specialty

Approved: 2 Semester Credits

Tom Lacksonen
Investigation Advisor

The Graduate School
University of Wisconsin – Stout
May 8, 2001

The Graduate School
University of Wisconsin – Stout
Menomonie, WI 54751

Abstract

Writer: Howard, Dominic A.

Title: Prioritization and Optimization of Software Deployment

Major: Management Technology Advisor: Tom Lacksonen May 2001 39pgs

Style Manual: APA 4th Edition

As technology continues to advance, companies are forced to keep up with these changes or lose their competitive advantage. However, there are significant costs also associated with the implementation or acquisition and installation of that technology. In order to make the most efficient use of the resources to gain the most benefit from that technology, it is important that these implementations are done in a cost effective manner. Many factors contribute to the cost of implementing or upgrading to new technology. Some of these are fixed costs, or have very little flexibility. Many are necessary in order to complete the project, no matter what other choices are made. Some choices, however, have a significant impact on the cost of implementation without providing any additional efficiency for that cost.

Decisions regarding the methods and mode of software deployment are one group with such an impact. Poor decisions, frequently made by managers with no Information

Technology (IT) experience, can significantly raise the cost of a deployment project. This study looks at the impact of allowing political prioritization to take precedence over IT based decisions. When these decision strategies were compared within a division of the University of Wisconsin – Stout, significant variations were discovered. This scenario led to a 38% cost overrun between the models, and 2 weeks of additional time to complete the project with no benefit to the division as a result.

Table of Contents

Chapter 1
Introduction

So much of business focus is on the bottom line, and how profitable the organization is, yet many organizations neglect the impact of Information Technology (IT) on their organizations. This is especially true when it comes to managing the costs of technology installations and upgrades. In most aspects of business, separation of powers is a wise principal, allowing for specialists in each facet of the operation to make decisions about what is best for the organization and their unit. In the case of Information Technology, this is frequently not the case.

Even very small changes to an IT strategy by managers not familiar with the implications, interactions, or scope of Information Technology can have a devastating impact on the bottom line of the organization. Known in chaos theory as the butterfly effect, the theory originated in the weather research of Edward Lorenz. In an attempt to simplify a weather simulation as it was being transferred between computers, Lorenz changed the number of decimal places involved in his data entry from 5 to 3, a seemingly insignificant change. The result was a completely new distribution of weather patterns in the simulation that bore almost no resemblance to the original projects (Gleick, 1987).

This study utilizes a spreadsheet model based on the operational and political structure of a division within the University of Wisconsin - Stout to assess and compare multiple methods of software distribution to assess the best approach in a specific hardware and political deployment environment. The division consists of 11 separate operational departments with a support requirement of 234 computers. This upgrade process involved bringing all of those computers to Windows2000© and Microsoft

Office© 2000 beginning with the second full week of January, 2001 and continuing as a single project until completion 8 weeks later.

Three models will be compared. The first model describes the actual process required by the division to process this software upgrade procedure. It is based on the political assignment of priority to departments based on the stature of their administrators without regard to the functionality or needs of the department. All departments must be completed in their entirety in the order dictated by the administrative concerns. The second approach is similar to the first, respecting the need to complete all units as a single entity, and in the same time period rather than break up the interoperability and coalescence of the departments. The third method is simply a best-fit model, allowing for the abandonment of all political and departmental boundaries in order to complete the process as quickly and efficiently as possible. All models are based on the same staff hours and availability, and the availability of the departments during their eligible slotting. All upgrades would be performed in the same manner, regardless of approach, so that the only variation is the degree of managerial interference.

Deployment methodologies for this project were evaluated using the Solver add-in for Microsoft Excel. Solver was set up to maximize the number of hours used in each evaluated week subject to not exceeding the total amount of hours available for that week. The solver model also has to check against weekly tables to determine which departments are available to be upgraded in a specific week, whether the department has sufficient priority to be considered for upgrade during that week, and that the department has not already been completed. The model also calculates the total cost of each week and the number of hours not utilized in completing the upgrade project based on the number and pay rate of the staff available for that particular week. The model also updates a completed table once a department has been chosen, so that it is not considered available for any future weeks. This process is repeated until there are no departments left to finish.

The use of this summary information is designed to provide a practical framework in which to test ideas, and the conclusions of this project will be implemented without any reference to the use of the optimization mode. The expectation of the simulations is that the greater the political interference involved in the system, the less efficient, longer duration, and more costly the process will be.

Chapter 2
Literature Review

Technology is evolving at a remarkable pace. Few would have predicted even 10 years ago that there would be such dramatic and significant changes to the face and application of computers and technology. Even those whose job it is to handle, adapt, and anticipate the role of technology are hard pressed to maintain knowledge of all aspects of these changes.

The physical hardware of computer systems evolves so fast that any new computer purchased is obsolete within months, and insufficient to deal with the newest hardware and software in only a couple of years. Internal system architecture changes so drastically in both design and speed that components such as memory and CPUs cannot be interchanged between generations and types of system boards.

User peripheral devices have marched through an endless variety of proprietary systems to evolve into a pseudo standard by which single generations of hardware exist to comply with a standard, but that standard evolves between generations, making it increasingly more difficult to continue to use existing hardware in new systems. Nowhere is this trend more obvious than in the configuration of the POST (power on self test) of the 1998 version of the Phoenix© BIOS (basic input output system, the part of the system board that allows the computer to boot) that shipped with several models of Gateway© computers. Any hardware device that the system was able to recognize during the boot cycle was labeled as a legacy device. By this model, all devices that work are outdated (legacy) and everything else is beta (not yet completely tested).

System speeds of 25 Megahertz were once considered blazing, while current systems are available at speeds to 1.5 Gigahertz (1500 Megahertz) and that trend will only increase. Systems that once contained 8 Megabytes (A million bytes of information) of memory and 500 Megabyte hard drives now come standard with 512 Megabytes of memory and drive space is measured in Gigabytes (a billion bytes of information) and Terabytes (a trillion bytes of information). For sake of comparison, an average novel is about 1 Megabyte, and the average e-mail if 10 kilobytes (.01 Megabytes).

What is driving this amazing increase in speed and storage is the software utilizing these machines. In the early days of personal computing, the disk operating system (which allowed the computer to run), a word processing application, and a user's data files could all be stored on a single 360-Kilobyte floppy disk. This disk is a fourth the capacity of the floppy disks utilized today, and even this technology has been enhanced to provide larger storage. Imation© produces the LS-120, which is the same physical size as a floppy, but hold 120 Megabytes of data. Iomega© produces its Zip Disks in 100 and 250 Megabyte formats in a disk that is only slightly thicker than a floppy, and there are other formats that allow the storage of a Gigabytes of data in the same form factor as the Zip Disk.

A typical installation of Microsoft MS-DOS 6.22 and Windows 3.11 shipped to the end user on less than 8 floppy disks, and could be installed in 5 or 6 Megabytes by a professional user. An early version of Microsoft Office (95) shipped on 4 floppies. In contrast, the current versions of these products are no longer even shipped on floppy disks, and instead span multiple compact disks. Windows 2000 Professional, Microsoft's current home user operating software takes 250 Megabytes for a typical installation, and Office 2000 Professional can take in excess of 450 Megabytes for an installation. In this scenario, it should be obvious why the speed and capabilities of computers has had to increase in order to support this phenomenal increase in the size of the software needed just to run the computer.

The Internet has also been a driving force behind the advances in storage capacity. Both the amount of data being supplied to users and the amount of data that users wish to store from Internet sources requires large repositories. Collaboration of users and sharing of processes and applications across a network has also increased the speed requirements of computers. In the client/server environment, the server systems must be as powerful as possible to facilitate the number of requests coming for all of its users in a timely fashion.

Other driving forces in the evolution of computers lie beyond the scope of this discussion, as they have no practical value to the business operation of computer systems. This area includes, and is lead by, games. The quest by game companies to provide more realistic and adaptable games requires more processing power for this to occur in real time for the user. Addition factors include multimedia entertainment (Internet radio, streaming media files, movies, and music) and the trade in both legal and illegal software and data. While these factors may have an impact on the cost of maintaining a network, they are not directly related to the computer.

In order to achieve the biggest benefits from these developments in technology without draining the bank, it is necessary in medium to large-scale organizations to rely on an IT (Information Technology) specialist or department. This group is also responsible to maintaining a floor level of technology to ensure that all users in an organization are able to use the same software and processes in order to be able to communicate and share information effectively. These services can either be employees of the company, or personnel who are brought in as consultants or as part of a service agreement with the hardware or software provider or reseller. This paper deals with organizations that have a dedicated IT staff to facilitate the technology needs of the operation.

The primary motivation for using skilled specialists to evaluate, maintain, and control a computer system is financial. With the rate at which technology changes, there is a fine line between being too far behind to make effective use of your personnel, and the excessive costs of being on the bleeding edge of technological improvement. Timing and knowledge are crucial in making the most cost-effective decisions. Among the factors to be considered in this equation are the total cost of ownership of the system (TCO), return on investment (ROI), the advantages of standardization and automation, and the level of autonomy afforded the IT department.

In order to consider and use TCO as a basis of decision making for IT strategy, it is necessary to understand the fundamentals and flaws of the system. Originally conceived and quantified by the Gartner Consulting Group in 1987, TCO was designed to be a tool to allow administrators to determine the impact of technology decisions. It provided a method to understand how the process affected the amount of money required to support a user and their applications as well as the hardware (Gartner, 1997). Gartner's biggest discovery during this process was that the physical cost of the hardware was an insignificant portion of the TCO.

Since its introduction, many different companies and reviewers have evaluated TCO, and the consensus decision is to include factors in several dimensions. The four main factors are budgeted costs: hardware and software, network management and technical support, development of applications and processes, and communications costs for network connectivity. The indirect costs included the damage caused when a user attempted to fix their own problems and the cost to the business of downtime (Gartner, 199; Minasi, 1998; Ivens, 1999). In 1997, Gartner officially revised these elements, and began work on the implementation of a revised TCO model to account for a more realistic interpretation of how office processes interact and flow (Gartner, 1997). The new model also looks at the impact of the skills required of a worker to do their job, and the amount of processing power required to do that job efficiently.

TCO is not a perfect science, and is usually treated as more of a guideline than a science, but it is a viable option for estimating costs and savings for an operation (Milligan, 1999; Gartenberg, 2000). Some critics also indicate that the model does not take into account enough of the soft costs involved in the use of a network to interconnect devices, particularly between offices and geographical regions (Liebmann, 1999). This aspect seems especially troubling to purchasing specialists who lack detailed knowledge of networking systems (Milligan, 1999).

To compensate for these issues, TCO is often used in conjunction with ROI calculations to determine the gross benefit to the company for the effort to improve it systems. This method adds a checksum to the calculations and provides another yardstick for comparing effectiveness of options. However, ROI calculations are much more easily and accurately performed on a real situation, rather than being projected onto a future idea. This is especially true if the method is a prototype and there is little comparative data (Liebmann, 1998; Milligan 1999).

Despite these shortcomings, TCO remains the most viable method of evaluating options (Minasi, 1998). This is especially true of the revised TCO model created by Gartner in 1997 as part of a large-scale study involving 20+ large international high technology companies (Gartner, 1997).

Once aware of the limitations of the model, how best is it used? As long as the same set of parameters are used for each alternative in the comparisons, the results are at least acceptable in determining which alternatives are better than others, and with practice, the results can be made reasonably accurate as well (Cappuccio, Kirwin, Pawlick, and Namasivayam, 1996). This approach compared the best practices model with the actual results at the completion of the project. Cappuccio, Keyworth, and Kirwin (1996) also demonstrated the effectiveness of this model by including life-cycle calculations for hardware, and achieved similar results.

Other interpretations including only the costs that can be easily and physically quantified have also been tested. While these approaches make the accountants happy, they often fail to reliably predict the full extent of cost saving (Workforce, 2000). The strictly hardware comparison approach yields even less accurate data, and is useful primarily in determining the start-up cost of a new project rather than the integration of an existing issue (Kapoor, 1999). As a result of the comparison of these methods, this project will use the soft cost plus hard cost model to estimate project costs.

No matter what method is used to calculate the TCO of the technology of an operation, it is important that a consistent approach is taken, and more importantly, the

TCO be reduced as far as possible to aid the overall profitability of the operation. In order to achieve this goal, there are three main issues to be addressed. These issues are the standardization of hardware and software platforms to reduce support and maintenance costs, the automation of software delivery, and the autonomous functioning of the IT department within an organization.

The first and most obvious place to start standardization is with the PC hardware (Avery, 1999), and while this is an admirable goal, it lacks practicality in anything but a small-scale operation. Technology is changing at an ever-accelerating rate, with life cycles that were 4 to 6 years during the last PC generation already plunging to 3 years; and this tend is likely to continue. In some cases products whose purchases have been planned are no longer available at the end of the budget approval period.

In addition, the capital required to maintain complete hardware compliance would put undue strain on the budget process in a single year, pushing other projects away or tying up capital during the intervening years to prepare for the large purchases. Differences in purchasing contracts and product reliability cycles also impact what technology is available from year to year and order to order. The only practical way of purchasing in a large organization is to stagger the purchases across the life cycle of the hardware, making exceptions for those users or departments who have a shorter life cycle.

The best you can then hope for on hardware is a pseudo-standard based on the evolution of technology and the principle that technology needs are best met by placing the most powerful equipment where it is needed, and allowing the older technology to filter down through the levels where it is still useful before being removed from the operation. The best solution under these conditions is to concentrate on the standardization of the software components of the operation.

While software is also constantly evolving and changing, it is far more flexible in allowing for the use of a wide variety of hardware options, and is generally on a longer development and obsolescence cycle than is hardware. In addition, there is greater vendor support for both upgrade paths and tools for the standardization of baseline deployments of the software (Edwards, 1999; Smith, 1999).

The use of a baseline deployment lowers the TCO for computers by giving the technical support operation a baseline from which to operate. This provides a set of known standards and parameters that are consistent in all machines, and if problems occur, they are the result of user changes to the system. This allows for the support staff to track these changes through either monitoring the recent changes to the system using internal auditing feature, or the use of external tools capable of integrating and logging the actions of the user (Mutek, 2001). In addition, there are software tools that will prevent the user from changing any of these parameters, thus preventing the problems from occurring in the first place (Minasi, 1998; Smith, 1999).

In addition to support at the level of the desktop, there are many alternatives to allow for the management of a network workgroup or domain through policy restrictions and user identification. Such processes can isolate issues and modify software availability down to the user and computer from which the application needs to be run. In this way, a user of a computer has access to all of the resources that they need, but other users of the same computer do not. This approach prevents users tampering with information and software not intended for them to use in the performance of the job (Pierce, 1999; Minasi, 1998; Smith, 1999)

In addition to the standardization or baseline deployment software and the ability to control user interference with its proper operation, significant improvement in TCO reduction can occur through the use of automated processes of software deployment. This can affect anything from new installation to repair or replacement of software or the distribution of upgraded versions (Minasi, 1998; Edwards, 1999). IBM's LAN Manager and Microsoft's Systems Management Server and Windows 2000 Active Directory are current examples of integrated software that allow for the automated distribution of software in an active phase. This means that changes can be made to an existing installation, rather than destroying and upgrading the software library of a client computer.

In the faster, but destructive mode of operation are products like Drive Image from PowerQuest© and Ghost from Norton© software. Both of these products rely on the destruction of all existing software on their target computers, replacing them with a completely new image. This method allows for faster installs than individual distribution methods, and has support for network-based, 'boot from floppy and go' approaches, and also supports the use of simulcast technology to perform installs on a group of clients at a single time.

The biggest pitfall to this approach is the need for a constant, reliable, high seed connection. This approach usually works well in a single geographic location served by an isolated network segment, but will put excessive stress in a leased, low-speed line that is commonly used to connect remote locations (Williams and Robinson, 1999; Zhou, 1999). Because these technologies rely on the destruction of existing data, the network link must not fail during the installation process, or all systems on the link will be inoperable.

There are many options for avoiding these problems, as long as they can be planned in advance. These may range from high cost, redundant network paths with fail-over operations to the use of multiple servers or the addition of flexible technology, such as wireless connectivity (Anderson, 2000). Among the more cost effective means for handling remote locations is to either use a single distribution preparation facility in which all incoming computer requests are directed, or the release of the distribution media to remote sites.

The single distribution point has a higher cost in that all hardware must be shipped twice, but ensures that the same versions of images are used, and that quality control of the set-up process is easily managed. This method is generally preferred in small companies with branch offices that cannot afford to house their own IT staff. This may be used if the IT staff in a remote location is not sufficiently large enough to handle both the normal workload, and the demands of an automated installation.

In operations with separate IT departments in remote sites, the distribution media can be replicated and sent to these sites. Since the existing staff is in place, there is no need to shift personnel to cover the process, and there are no additional shipping costs involved for the hardware. Care must be taken to ensure that all branch offices are running the same version of the images created at the main office, and training of the users at the remote sites becomes far more important. If the installation at these sites fails, not only is the original time wasted, but also the entire process must be run again, usually involving sending technicians from the main office to the remote site. Also factored into the cost is any productivity lost due to the unavailability of the hardware until the problem is resolved. If such a scenario must be used, it is recommended that the operation be run in stages, so that a critical failure of the system will not render the office dead.

While standardization is an important part of the TCO management equation, understanding the life cycle of the software is also in critical to understanding the implications of this approach. Software changes like hardware, adding features and components, becoming more useful to the end user. At least that is what we would like to believe. In order to ensure that we are providing the best service to our users without sacrificing performance or overpaying, the life cycle must be studied.

Software that is outdated loses efficiency and may cause problems in support as a legacy product. This is quickly becoming evident with DOS applications as Windows 2000 begins to grab market share. On the other extreme, the newest software often requires the newest hardware and technology to support it, and that adds significant costs. Ideally implementation of a new or revised product would occur early in the cycle, but not early enough to be exposed to the bugs and problems of the software (Ambler, 1999; Hall, Heimberger, and Wolf, 1998).

The use of agent, software capable of determining the limits of existing hardware configurations, and deciding which clients are best suited for automatic updating also acts to lower TCO. The goal of these agents as they gather information about the clients in a network is to provide feedback to the IT department as to who is currently capable of following an execution path. In addition, the agents can recommend changes to the system configuration that would enable the system to proceed, and will also be able to identify hardware systems beyond the possible range of updating in a cost effective manner.

The use of agent technology allows the decisions about enhancing software to be decentralized. Agents can replace IT staff in remote locations to decide on a course of action, and are able to follow that course alone, if necessary. Such methods free the IT staff to focus on proactive measures regarding the network and hardware, as well as software, environment in which they operate (Ambler, 1999; Hall, Heimberger, and Wolf, 1998; Hall, 1997; Ambler, 1999; Hall, Heimberger, and Wolf, 1999). To aid in the efficiency and generalizability of this approach, it has been suggested to that a specific programming language and syntax be proposed to support this effort, allowing for cross platform predictability n the engineering of these distribution plans (Hall, Heimberger, and Wolf, 1999).

The end goal of all of this is to minimize TCO in operating a network-based environment, and the best method of achieving this is through predictable control. If the IT staff is able to control the distribution methods and parameters of its software components, the costs of support go down. Control of hardware and use access to change it lowers the cost and frequency of technical support. Delivering the right software to the right people in the right place at the right time increases productivity, improving the return on investment that a company places into the evolution of its IT operation and those operations dependent on technology to drive their efficiency. Simply put, the best way to manage costs and increase return on investment is systemic control (Hoek, Hall, Carzaniga, Heimberger, and Wolf, 1998; Hall, Heimberger, and Wolf, 1999).

In order to ensure that all of these disparate elements function correctly, and are providing the best return for an operation, it is best that experts in the use of these technologies run them. While the accountants and officers of the operation feel the need to be involved, the best outcome to achieve lower TCO and increased ROI on technology is to trust that process to the IT staff. Because of the daily exposure to the process, and their access to other exerts in the field of technology, they are best suited for the job Wiegers, 2001a; Gottesdiener, 2001).

This is not to imply that there should be no oversight of the IT department, as it is the nature of technology advancement that the IT staff will have a different view of the vision for technology within a company. Therefore, it is still the responsibility of the executives of the operation to develop the vision and principles that drive the process. Only within those guidelines should IT be allowed to proceed. Once these parameters have been established, it is then time to let the IT staff work on what it is most familiar with (Wiegers, 2001b). In proactive companies, the general philosophy is that the IT staff is paid to provide its expertise to the company and the responsibility of the executives is to supply vision that guides these principles. This is true in the case in which this scenario is being applied, as confirmed during a recent interview with the divisional executive director (Johnson, 2001). Allowing too much interference or influence to be exercised by a well meaning or power hungry manager can have a significant effect, as detailed in the so called “butterfly effect”, the basis on chaos theory (Glieck, 1987). Simply put, the flapping of a butterfly’s wings in Peking can have a significant effect on the weather in New York City a month later and no one will ever really know why or how this impact occurred.

Under these parameters, this project is designed to examine the difference in TCO to upgrade an existing organizational software infrastructure based on three different models. The first model allows the IT department to operate on a sliding scale, crossing departmental boundaries, if necessary, to achieve the most efficient rate of installation. The second model is a modification to the first, but insisting that all departmental boundaries be respected, but no political influences are used to determine the order of service (Johnson, 2001). In the final model, all of the existing political and departmental constraints are applied as determined by the existing hierarchy within the division, including the scheduled availability of departments being inflexible in regard to the entire process path. These comparisons will allow for a partial cost estimate to be made on the impact of non-technical management interference with the IT operational strategy, and allow for methods to be developed to minimize that cost by suggesting changes to the oversight structure.

Chapter 3
Methodology

The model for this process is based on three methods of bringing various software levels to a single standard, based on the hardware capabilities of the existing installation, and the political constraints governing the level of control exercised during the upgrade operation. These options are install new computer, perform clean install on existing system, and upgrade existing system. The timeframe for each one of these installation options is the average time of tests done in a deployment lab using the median PC configuration for each type of installation. All of these options assume that the computer being updated is capable of running the new software, as defined by the manufacturer's minimum recommended system specifications.

The first option is the ideal scenario. However, due to budgetary constraints and the need for cyclical replacement schedules, it is not the only procedure employed. This scenario consists of the delivery of a new computer, and the ability to start completely from scratch and specifications with this system. The hardware substantially exceeds the manufacturer's requirements, and there is no pre-existing software to deal with. All user data will be transferred in a single pass at the end of the installation process. This option requires approximately 2 hours per machine to complete (Table 1, process 1). The second option, while not as efficient as the first, involves the cleaning and reinstallation of software on an existing system. This option takes longer than the first because the hardware is older and slower because of its position in the replacement cycle. In addition, all of the user data must be transferred to network storage or other backup process before the operation can begin. This option also requires a thorough check of the hard drives involved to ensure that no complications develop in mid process. Once these provisions have been made, the process is similar to the one used in the first option. Testing indicates a mean time to install of just over 3 hours for this process (Table 1, process 2).

In both of these scenarios, installation time and complexity is reduced through the use of an install image. This image is compressed, and already contains all of the software required by the user to perform their work. Once this image is copied to the target computer, it goes through a hardware detection phase to customize itself to the idiosyncratic distribution of hardware, then all other components function normally. This process can be accomplished through the use of CD-ROM images, or distribution across a network installation point. In either case, the user intervention and provision of data to complete the setup is almost completely eliminated, allowing for the process to be standardized. This methodology also makes troubleshooting easier, as all platforms start from the same basis.

The third option is far more complex and time consuming. Again, it involves older, slow hardware, but in addition to the speed issues, the existing software must be preserved and updated rather than replaced. This scenario involves either mission critical systems that cannot be shut down for a significant length of time, systems where there is no original installation media available for specialized software, or where the user has sufficient political clout to not have to comply with the clean and redo methodology.

This option requires backing up both the user data and a recovery image of the system to the network to allow for the system to be rolled back to its original state in the event of an emergency. All software on the computer must be individually upgraded and service packs applied instead of the use of the pre-compiled images. In addition, all hardware consistency checks must be done in a non-destructive manner. The result of these limitations is a set of installations that take over 5 hours each to accomplish (Table 1, process 3). Each department contained a mix of multiple installation types that had to be performed (Table 2).

These different methodologies are required across all three of the deployment scheduling options, and cannot be modified to aid in the deployment planning process. Within these conditions, three options of deployment scheduling will be addressed. The first option, which was the one that was required by the organization, involves following a specific set of political guidelines for order of installation, and the maintenance of departmental equivalence in software. In other words, departments must be completed as a single unit to retain interoperability. The second option retains the continuity of the departmental configuration, but ignores the political ordering of the process in favor of a best-fit model. The final option is the optimization of available resource to complete the entire project as quickly as possible with the staff available, and without regard for departmental continuity or political definitions.

All options rely on the existing staffing conditions of availability (Table 5) and available hours (Table 6), as well as pay rates (Table 6) already in place within the organization. Table 5 is a binary matrix of staff availability. A number on the left represents each staff member, and the columns represent the weeks. A “1” means the staff member is available for all hours normally scheduled for a week. A “0” means that the staff member is not available for that week. Table 6 is a sum product of table 5 and the normal work hour distribution of the staff. Total hours is a sum total of each weekly column.

The cost factor here assume that the staff assigned to work on the software distribution project is unable to work on other projects during this time, and the project concludes at the end of the final installation. In addition, the downtime blocks required of specific departments are those of the organization (Table 7). These exist due to various programming issues, pre-planned events, and scheduled disruptions to the operations caused by external events, but that also preclude access to the hardware in a useable manner, such as maintenance of the facility or painting. TCO comparisons are based on the discrepancies between each of these models.

The objective of the model to was to maximize the number of labor hours used in any particular week as long as it did not exceed the number of hours available from the staff for that week. This maximum was determined by allowing solver to attempt combinations of departments to find the largest number of hours that could be used, subject to the availability conditions of the departments. These conditions included the fact that each department could only be selected once, and is then removed from eligibility regardless of any other favorable conditions. Another factor that is considered is the political availability of the department. This factor is constrained only in option 3, and requires that all departments of higher political priority be completed before allowing any other departments to be considered available. All solver decisions to include or exclude a department that was available were binary, either completely used or not at all, so that the solver model did not attempt to use partial departments to improve its efficiency.

The table on the right of Figure 1, O1 to AA21, is the departmental availability. A “1” means that the department is available that week. Department identification numbers are defined in row 1. Column “P” denotes the week of the project. Figure 1, Row 26 denotes the administrative priority levels used by option 3 of the model. Figure 1, A1 to M21 is the availability matrix for the department. It is derived from the multiplication of weekly availability (Figure 1, O1 – AA21), usability (Figure 1, Row 29 [priority {Row 26} = phase {O33} for a “1”]) times the inverse of previous selection (Table 3 or 4 depending on option).

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	AA
1	Dept		1	2	3	4	5	6	7	8	9	10	11		Dept		1	2	3	4	5	6	7	8	9	10	11
2	week	1	0	0	0	0	0	0	0	0	0	0	0		week	1	1	1	1	1	0	0	0	1	1	1	1
3	week	2	0	0	0	0	0	0	0	0	0	0	0		week	2	0	0	0	0	1	0	1	0	1	0	1
4	week	3	0	0	0	0	0	0	0	0	0	0	0		week	3	1	1	0	0	0	1	1	1	0	0	1
5	week	4	0	0	0	0	0	0	0	0	0	0	0		week	4	1	0	1	1	0	0	0	0	0	0	1
6	week	5	0	0	0	0	0	1	0	1	0	0	0		week	5	1	1	1	1	1	1	0	1	0	1	1
7	week	6	0	0	0	0	0	0	0	0	0	0	0		week	6	1	1	1	1	1	1	1	1	1	1	1
8	week	7	0	0	0	0	0	0	0	0	0	0	0		week	7	1	1	1	0	1	1	0	1	1	1	1
9	week	8	0	0	0	0	0	0	0	0	0	0	0		week	8	1	1	1	1	0	0	0	0	1	1	1
10	week	9	0	0	0	0	0	0	0	0	0	0	0		week	9	1	0	0	0	1	1	1	1	0	1	1
11	week	10	0	0	0	0	0	0	0	0	0	0	0		week	10	1	1	1	1	1	1	0	0	1	0	1
12	week	11	0	0	0	0	0	0	0	0	0	0	0		week	11	1	1	1	1	1	1	0	0	1	0	1
13	week	12	0	0	0	0	0	0	0	0	0	0	0		week	12	1	1	1	1	1	1	0	0	1	0	1
14	week	13	0	0	0	0	0	0	0	0	0	0	0		week	13	1	1	1	1	1	1	0	0	1	0	1
15	week	14	0	0	0	0	0	0	0	0	0	0	0		week	14	1	1	1	1	1	1	0	0	1	0	1
16	week	15	0	0	0	0	0	0	0	0	0	0	0		week	15	1	1	1	1	1	1	0	0	1	0	1
17	week	16	0	0	0	0	0	0	0	0	0	0	0		week	16	1	1	1	1	1	1	0	0	1	0	1
18	week	17	0	0	0	0	0	0	0	0	0	0	0		week	17	1	1	1	1	1	1	0	0	1	0	1
19	week	18	0	0	0	0	0	0	0	0	0	0	0		week	18	1	1	1	1	1	1	0	0	1	0	1
20	week	19	0	0	0	0	0	0	0	0	0	0	0		week	19	0	0	1	1	1	0	1	1	1	0	0
21	week	20	0	0	0	0	0	0	0	0	0	0	0		week	20	1	1	1	1	1	1	0	0	1	0	1
22	Install		0	0	0	0	0	1	0	1	0	0	0														
23	Demand		0	0	0	0	0	82	0	47	0	0	0	129	Hours	<=											
24														133	threshhold												
25	name		1	2	3	4	5	6	7	8	9	10	11														
26	priority		1	3	1	3	3	2	1	3	2	3	2														
27	size		39	22	18	19	21	25	39	15	18	13	5		234												
28	procs		126	64	57	67	73	82	105	47	50	36	21		728												
29	Usable		0	1	0	0	0	1	0	1	0	1	0														
30																											
31	total cos	staff	hours	avail	cost																						
32	1000	1	40	1	25									week	5												
33	360	2	20	1	18									phase	2												
34	450	3	30	1	15																						
35	0	4	20	0	13							3 approaches															
36	270	5	18	1	15						1	raw hours															
37	450	6	25	1	18						2	best fit															
38	2530		133								3	admin prioritie															
39			19.02																								

Figure 1 Solver Compilation Matrix

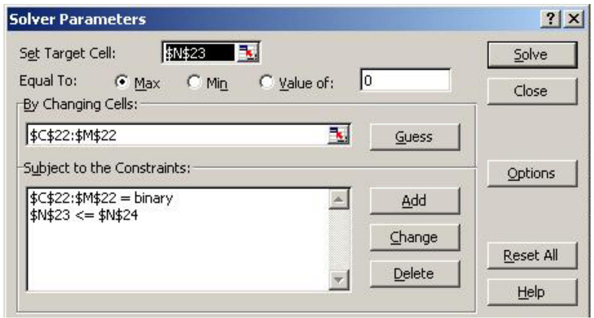

Figure 2 Solver Constraints

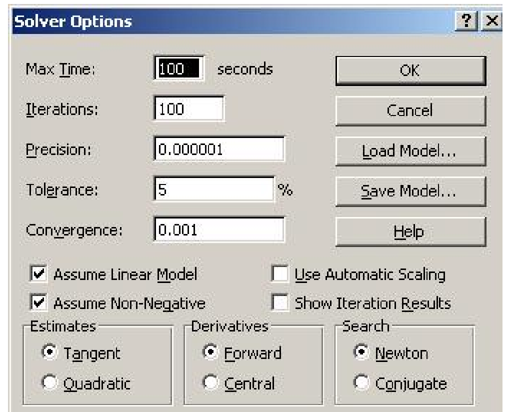

Figure 3 Solver Limits

The Objective (Figure 2, Target Cell) of total hours (Figure 1, N23) is a sum of demand (Figure 1, Row 23), which is determined by the sum product of install (Figure 1, Row 22) and procs (number of procedure hours required for the department [Figure 1, Row 28]). The maximum hours allowed (Figure 1, N24) is equal to the hours available for the week (Figure 1, C38). Total demand must be less than or equal to that threshold (Figure 2, Constraints). The solver configuration screens are shown here, showing how excel is configured to follow these parameters.

Figure 3 shows that Solver is allowed to assume non-negative values because the change cells are restricted to binary values (Figure 2, Constraints), excluding any possibility of a negative number. Solver is also able to assume a linear model in this process, as all of the conditional logic in determining availability does not impact the testing of the options available during each weekly run, allowing for the use of an integer programming solution (Plane, 1996). This allows for faster processing of the equations and greater confidence in the accuracy of the results, and that they are they optimal solution to the equation, as explained in Plane (1996).

Chapter 4
Results, Conclusion, and Discussion

The Excel models were run for each scenario, applying the applicable rules for each option, until all of the departments had been completed. These rules varied between the modes, with option 2 only requiring the completion of any department started in a specific week, while option 3 also required that the political priority of each department also be considered. The only constraint of the first option was that the total number of available hours not be exceeded during the scheduling runs.

The impact of each option was evaluated using the variations in the cost of staff hours required to complete each run. All other costs associated with these options are fixed regardless of approach. Cost models were compared using total costs, wasted hours, and waste cost. Total hours are the total number of hours that were available for a specific week. Waste hours are those hours between the maximum solution provided by solver and the total possible hours that could have been worked. Waste cost is the sum of the number of hours not utilized for the project and the average cost of those hours. Waste hours are not recorded for the final week of the project, as all staff members would be released to other projects, and a new cost scale.

As expected, placing no restrictions on the operational approach and prioritization schedule (option 1) does indeed yield the shortest project run, and the restriction to departmental political priorities (option 3) took the longest. Installation approaches using either no form, or limiting the operation to complete divisions without regard for political priorities were both completed during the sixth week of the project, and concluded 57 work hours apart (Table 8). This amounted to a 3 percent difference in cost between the two methods, or $438.50 (Table 9). The order of completion for method two is shown in Table 3. Due to the piecework nature of method 1, no completion matrix is available. Given the number of departments (11, Table 2), number of computers involved in the process (234, Table 2), and anticipated work hours (734, Table 2), the tight proximity of these results was not expected.

Method 3, the politically directed approach to software installation and distribution took the longest by a significant margin. This approach took 266 hours longer than method 2 and 323 hours longer than method 1. Even allowing for the inoperative week, it took 133 hours more than method 2 and 190 hours more than method 1. As is, this disparity amounts to a 38% (Table 8) increase in labor hours at a cost of $4892.76 (Table 9).

The difference between the times to complete on these two extremes was over two weeks, although there is a one-week lapse where the next department in priority structure was unavailable during its first week of accessibility on the political scale, exaggerating the differences. This occurred during week 3, when a single department had to be taken off line due to the return of students to campus, and the requirement of that department to facilitate the scheduling of meeting space for groups and organizations that could not be rescheduled, and required that the operation not be interfered with during that time. This effected only model 3, because of the structured requirements for order of completion. The other options were able to flex and use other departments during that specific week, avoiding the negative consequences of the scheduling conflict. Because this simulation is based on an actual operations timetable, this exception must be included in the calculations of time required.

In an environment as small as the one in this study, such a seemingly innocent decision to allow the interference of managerial techniques including the political prioritization of distribution methods for software has a significant effect. If such decisions can cause a 38% increase in cost in an organization in which the entire process involved only 11 departments with 234 computers and 6 support personnel, the impacts of these kinds of decisions can be costly to an organization with a significant number of computers. The complexities of this issue grow quickly. Even in a small environment such as Stout, this accounts for only about 9% of the total computers owned and supported by the University. For global corporations, such as Microsoft or General Mills, with offices worldwide, this number could easily increase 50 to 100 times.

Waste cost in these options is highly dependent on the size of the departments and the political priority of those departments. The larger the departments are, the less likely it is that there can be efficiency in scheduling the completion of each department if there is no option to split them. In addition, larger departments tend to have higher political priority, so if the model is based on that priority, and smaller departments that could have been added into consideration during option 2 are excluded from consideration in option 3. This combination can easily exacerbate costs, especially in the early stages of a project. The more large departments, the bigger the discrepancy will be, and the farther into the project it will last.

This project can only suggest general patterns of cost and efficiency, as there are several external factors that can impact the total numbers. Had the start date of the project, a political decision, been shifted forward a week (backwards was blocked by the holiday season) the loss of all of week 3 in option 3 would not have occurred, decreasing the gap between the options. Also, in large companies, or those whose priority is speed of transition, it is possible to involve the use of temporary employees, or to reassign staff members from other departments to help with the project. Also, arbitrarily breaking departments into smaller chunks and creating subdivisions with different political priorities would ease the level of inefficiency inherent in the large department model. In the frequently politically charged arena in which these decisions are made, it is unlikely that this option would be considered.

While companies claim that their biggest concern is the bottom line, and the responsibilities to the stakeholders of the organization, an issue such as a power play between divisions or administrators over political issues involved in an Information Technology project can have a significantly detrimental impact on that bottom line. While this does not mean that IT should be allowed to run free, it does show that by applying the knowledge of both software distribution and the time requirements of the processes involved in completing a significant upgrade proposition, the operational judgements of the IT department should be considered in the planning and development phase of such projects.

Additionally, as demonstrated by the variance in the amount of time required to perform different modes of software distribution and data handling functions, the expertise of the IT department should be sought in dealing with users or departments who feel that they are above compliance with the more efficient and optimal method of distribution. Table 2 demonstrates the 250% increase in time to completion of dealing with user demands versus the ability to manage all users in an identical fashion.

Magnified over 100 users, this will have a major impact in the cost and speed of completion for a project.

Though not covered in the scope of this project, there is also a significant cost saving in technical support by starting all users at a common point. If all users have identical configurations it allows for users to support each other instead of relying solely on the IT support infrastructure, and the users have less down time, as they do not have to wait for the IT response before being able to continue with their tasks. Also, if technical support is required, it can be provided more efficiently by using a base configuration, as it is easier to train support to deal with the variations from a single point rather than the complications of layered upgrades and software patches. There is also an impact in the hiring and training process for IT support staff. Training time can be reduced if there are fewer scenarios that the technicians are likely to encounter, and the skill level of the support personnel can be lower if there is lees need for intensive troubleshooting capability on the part of any individual technical support staff member.

Chapter 5
Summary

Solver was used to compare the available options for updating software within several university departments, according to the political restrictions and availability limitations of each model. As expected, having no operational restrictions on the prioritization schedule (option 1) yielded the shortest project run, and using political priorities (option 3) took the longest. Installation approaches using either no restrictions, or using any complete division (option 2) were both completed during the sixth week of the project, and concluded 57 work hours apart (Table 8), saving 3%, or $438.50 (Table 9). Option 3, the politically directed approach, took 266 hours longer than method 2 and 323 hours longer than method 1, a disparity of 38% (Table 8) in labor hours at a cost of $4892.76 (Table 9).

The differences between options 1 and 2 would have been more pronounced had the departments been larger, or the amount of support staff had been smaller, as it would have been more difficult to find an optimal solution within the constraints of the available hours. The ability to break operations into smaller units would aid the second option, but would have little impact on option 3, as these gains are overridden by the political constraints. This would also have no impact on option 1, as it ignores that size of the departments during the calculations.

At this point, the effects of the butterfly are making themselves clear, and no matter the rationale behind poor decisions, they are still poor decisions. The need to maintain political environments and hierarchy for its own sake is frequently at odds with the stated goal of management. While all layers of management pay lip service to the bottom line, most are willing to impact that number for their own personal gain. Compounding this is the constantly evolving nature of technology and the rapid demands for updated hardware and software. In most cases, it is necessary for the survival and profitability of the organization to keep up with the newest developments as they occur.

However, this challenge also comes with a price, measured in physical costs, such as hardware and software, personnel costs, and support costs. Managing this trinity is a touchy process that has yet to be clearly defined. As a result, if is almost impossible for managers with little or no active involvement in technology to make informed decisions regarding the installation and distribution of new technology. It makes sense from an economic and efficiency standpoint to rely on those who work with the issues on a continual basis, yet most organizations treat their IT departments as a drain on cash resources and as being overly protective of hardware and demanding of users. What few see to realize is that this is often in the best interest of the organization.

Since software is upgraded more frequently than hardware, this study looked at the impact of politicization on the cost and efficiency of software distribution, but the impact on hardware is similar. In the example provided, the choice by managers protecting what they believe are their own best interests cost them a 38% penalty in both cost of service and time to complete. What they also do not see is the cost in productivity when their staff members are off line because of their decisions to allow users to resist the methods designed by IT to maximize the efficiency of installation. They also fail to realize that delaying the installation of necessary systems makes both the staff and operations inefficient as they are not working at full capacity, and may not have access to information they need to enhance their job performance.

Information Technology pervades all aspects of the modern company, and its impact effects everyone. By continuing to allow the IT decisions to be changed by staff not qualified to judge those impacts, the organization is opening the doors to lost profits and inefficiency. Only when companies accept the fact that some autonomy must be granted to the IT staff to work in the best interests of the organization without regard for the naïve political boundaries and infighting that normally dogs such projects will technology even become completely integrated and seamless. It is also the only way for an organization to maintain its edge in technology without being sucked into costly spirals due to poorly planned upgrades. This is a definite case where being good in one area of management is not a guarantee of success in another, and smart managers will allow that realm to be handled by experts in the IT field.

Bibliography

Ambler, S. (1999, November). Thinking objectively: Effective software deployment, Software Development Magazine [On-line]. Available: http://www.sdmagazine.com/articles/1999/9911/9911o/9911o.htm

Anderson, C. J. (2000, August). Assure availability for the digital nervous system Communications News, 37(8) p26-27

Avery, S. (1999, September 16). Rockwell slashes costs by standardizing on PCs, Purchasing 127(4) p93-94

Cappuccio, D., Keyworth, B., and Kirwin, W. (1996) Total cost of ownership: The impact of system management tools [On-line]. Available: http://www.gartner.com

Cappuccio, D., Kirwin, W., and Namasivayam, S. (1996) Total cost of ownership: Reducing PC/LAN costs in the enterprise [On-line]. Available: http://www.gartner.com

Earon, S. A. (1998, November). The Economics of Deploying Videoconferencing. Business Communications Review 28(11) p43-48

Edwards, M. J. (1999, April 1). Microsoft's new sysprep tool, Windows2000 Magazine [On-line]. Available: http://www.win2000mag.com/articles/articles/Print.cfm?ArticleID=5047

Gartenberg, M. (2000, October 30). Myths behind TCO. Computerworld, 34(44) 52

Gartner Group. (1997) TCO analyst: A white paper on GartnerGroup's next generation total cost of ownership methodology [On-line]. Available: http://www.gartner.com

Glieck, James. (1987) Chaos: Making a New Science, New York: Penguin.

Gottesdiener, E. (2001, January). Decide how to decide, Software Development Magazine [On-line]. Available: http://www.sdmagazine.com/articles/2001/0101/0101h/0101h.htm

Hall, R. S. (1997). Software deployment in a wide-area setting using a distributed agent-based architecture. Unpublished master's thesis, University of Colorado, Boulder.

Hall, R. S., Heimbigner, D., and Wolf, A. L. (1999) Specifying the deployable software description format in XML [On-line]. Available: http://www.cs.colorado.edu/users/serl/cm/Papers/Dock/CU-SERL-207-99.pdf

Hall, R. S., Heimbigner, D., and Wolf, A. L. (1999) A cooperative approach to support software deployment using the software dock [On-line]. Available: http://www.cs.colorado.edu/users/serl/cm/Papers/Dock/dock_icse_150599.pdf

Hall, R. S., Heimbigner, D., and Wolf, A. L. (1999) Enterprise software deployment: It's the control, stupid [On-line]. Available: http://sern.ucalgary.ca/~maurer/ICSE99WS/Submissions/Hall/Hall.html

Ivens, K. (1999, September 13). Analyzing TCO as an industry tool, Windows2000 Magazine [On-line]. Available: http://www.win2000mag.com/articles/articles/Print.cfm?ArticleID=7190

Johnson, B. (Personal Communication, March 16, 2001)

Liebmann, L. (1998, November). Managing IT Investment: Less Isn't Always More. Business Communications Review 28(11) p52-55

Liebmann, L. (1999, January). The TCO myth. Communications News, 36(1) p92

Milligan, B. (1999, September 2). Tracking total cost of ownership proves elusive. Purchasing, 127(3) p22-23

Minasi, M. (1998, July 1). Just what is TCO, anyway, Windows2000 Magazine [On-line]. Available: http://www.win2000mag.com/articles/articles/Print.cfm?ArticleID=3565

Minasi, M. (1999, December 1). NT 5.0 TCO / ZAW update: Configuration and control management capabilities, Windows2000 Magazine [On-line]. Available: http://www.win2000mag.com/articles/articles/Print.cfm?ArticleID=4522

Mutek Software. (2001) Software black box flight recorders for client application support [On-line]. Available: http://www.mutek.com

Pierce, R. (1999, August 2). Flatten your NT domain structure, Windows2000 Magazine [On-line]. Available: http://www.win2000mag.com/articles/articles/Print.cfm?ArticleID=7051

Plane, Donald. R. (1996). Management Science: A Spreadsheet Approach For Windows, Danvers, Mass.: Boyd & Fraser.

Simplifying Network Deployments (March 1997). Communications News, 34 p25

Smith, M. (1999, December 1). Migrating to Windows 2000, Windows2000 Magazine [On-line]. Available: http://www.win2000mag.com/articles/articles/Print.cfm?ArticleID=7755

Tolly Group. (1997) Total cost of application ownership (TCA) [On-line]. Available: http://www.tolly.com

Van Der Hoek, A., Hall, R. S., Carzaniga, A., Heimbigner, D., and Wolf, A. L. (1998) Software development: Extending configuration management support in the field [On-line]. Available: http://stsc.hill.af.mil/crosstalk/1998/feb/deployment.asp

Wiegers, K. (2000, February). Stop promising miracles, Software Development Magazine [On-line]. Available: http://www.sdmagazine.com/articles/2000/0002/0002e/0002e.htm

Wiegers, K. (2000, May). Thinking objectively: Effective software deployment, Software Development Magazine [On-line]. Available: http://www.sdmagazine.com/articles/2000/0005/0005e/0005e.htm

Williams, W. and Robinson, P. B. (1999, July 19). What's it worth? Telephony 237(3) p36-39

WorkForce Software. (1997) Time and attendance white paper: Calculating return on investment of implementing a time and attendance product [On-line]. Available: http://www.workforcesoftware.com

Zhou, T. (1999, April 1). IP multicast and your network, Windows2000 Magazine [On-line]. Available: http://www.win2000mag.com/articles/articles/Print.cfm?ArticleID=5052

List of Tables

Tables

trial	proc 1	proc 2	proc 3
1	1:55	3:12	4:49
2	2:01	3:18	5:23
3	2:14	2:44	5:12
4	2:02	2:56	5:02
5	1:54	3:12	4:55
6	1:58	3:02	4:51
7	1:56	3:17	4:52
8	1:58	2:52	5:01
9	2:05	2:55	5:06
10	2:09	2:58	5:09
	2:01	3:02	5:02

Table 1 Average Install Time

Dept	*proc 1*	*proc 2*	*proc 3*	*Total*	*hrs*
1	15	12	12	39	126
2	12	5	5	22	64
3	9	3	6	18	57
4	6	5	8	19	67
5	8	4	9	21	73
6	9	8	8	25	82
7	20	15	4	39	105
8	4	8	3	15	47
9	8	8	2	18	50
10	9	1	3	13	36
11	0	2	3	5	21
				234	728
hours	*2*	*3*	*5*		

Table 2 Total Procedures by Department

Week	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*10*	*11*	*Dept*
1	1	-	-	-	-	-	-	-	-	-	1	
2	-	-	-	1	-	-	-	-	1	-	-	
3	-	-	1	-	1	-	-	-	-	-	-	
4	-	-	-	-	-	-	1	-	-	-	-	
5	-	-	-	-	-	1	-	1	-	-	-	
6	-	1	-	-	-	-	-	-	-	1	-	
method 2												

Table 3 Method 2 Installation Pattern

week	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*10*	*11*	*dept*
1	1	-	-	-	-	-	-	-	-	-	-	
2	-	-	-	-	-	-	1	-	-	-	-	
3	-	-	-	-	-	-	-	-	-	-	-	
4	-	-	1	-	-	-	-	-	-	-	1	
5	-	-	-	-	-	1	-	-	-	-	-	
6	-	-	-	-	-	-	-	1	1	1	-	
7	-	1	-	-	1	-	-	-	-	-	-	
8	-	-	-	1	-	-	-	-	-	-	-	

method 3

Table 4 Method 3 Installation Pattern

staff	1	2	3	4	5	6	7	8	9	10
1	1	1	1	1	1	1	1	0	0	0
2	1	1	1	1	0	0	1	1	1	1
3	1	0	1	1	1	1	1	1	1	1
4	1	1	0	1	1	1	1	1	1	1
5	1	1	1	0	1	1	1	1	0	1
6	1	1	1	0	1	1	1	1	0	1
	week									

Table 5 Staff Availability Matrix

Staff	**Pay**	Hours									
1	$25	40	40	40	40	40	40	40	0	0	0
2	$18	20	20	20	20	0	0	20	20	20	20
3	$15	30	0	30	30	30	30	30	30	30	30
4	$13	20	20	0	20	20	20	20	20	20	20
5	$15	18	18	18	0	18	18	18	18	0	18
6	$18	25	25	25	0	25	25	25	25	0	25
Total Hours		153	123	133	110	133	133	153	113	70	113

Table 6 Staff Hours Matrix

Dept		1	2	3	4	5	6	7	8	9	10	11
week	1	**1**	**1**	**1**	**1**	**0**	**0**	**0**	**1**	**1**	**1**	**1**
week	2	**0**	**0**	**0**	**0**	**1**	**0**	**1**	**0**	**1**	**0**	**1**
week	3	**1**	**1**	**0**	**0**	**0**	**1**	**1**	**1**	**0**	**0**	**1**
week	4	**1**	**0**	**1**	**1**	**0**	**0**	**0**	**0**	**0**	**0**	**1**
week	5	**1**	**1**	**1**	**1**	**1**	**1**	**0**	**0**	**0**	**1**	**1**
week	6	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**
week	7	**1**	**1**	**1**	**0**	**1**	**1**	**0**	**1**	**1**	**1**	**1**
week	8	**1**	**1**	**1**	**1**	**0**	**0**	**0**	**0**	**1**	**1**	**1**
week	9	**1**	**0**	**0**	**0**	**1**	**1**	**1**	**1**	**0**	**1**	**1**
week	10	**1**	**1**	**1**	**1**	**1**	**1**	**0**	**0**	**1**	**0**	**1**
week	11	**1**	**1**	**1**	**1**	**1**	**1**	**0**	**0**	**1**	**0**	**1**
week	12	**1**	**1**	**1**	**1**	**1**	**1**	**0**	**0**	**1**	**0**	**1**
week	13	**1**	**1**	**1**	**1**	**1**	**1**	**0**	**0**	**1**	**0**	**1**
week	14	**1**	**1**	**1**	**1**	**1**	**1**	**0**	**0**	**1**	**0**	**1**
week	15	**1**	**1**	**1**	**1**	**1**	**1**	**0**	**0**	**1**	**0**	**1**
week	16	**1**	**1**	**1**	**1**	**1**	**1**	**0**	**0**	**1**	**0**	**1**
week	17	**1**	**1**	**1**	**1**	**1**	**1**	**0**	**0**	**1**	**0**	**1**
week	18	**1**	**1**	**1**	**1**	**1**	**1**	**0**	**0**	**1**	**0**	**1**
week	19	**0**	**0**	**1**	**1**	**1**	**0**	**1**	**1**	**1**	**0**	**0**
week	20	**1**	**1**	**1**	**1**	**1**	**1**	**0**	**0**	**1**	**0**	**1**

Table 7 Departmental Availability Matrix

Approach 1	Week 1	Week 2	Week 3	Week 4	Week 5	Week 6	Week 7	Week 8	Total	%
Hours Utilized	153	123	133	110	133	76			728	
Waste Hours	0	0	0	0	0	0			0	0%
Total Cost	$ 2,790	$ 2,340	$ 2,530	$ 2,070	$ 2,430	$ 1,389			728	
Waste Cost	$ -	$ -	$ -	$ -	$ -	$ -				
Approach 2	Week 1	Week 2	Week 3	Week 4	Week 5	Week 6	Week 7	Week 8		
Hours Utilized	147	117	130	105	129	100			728	
Waste Hours	6	6	3	5	4	33			24	3%
Total Cost	$ 2,790	$ 2,340	$ 2,530	$ 2,070	$ 2,430	$ 1,827			785	
Waste Cost	$ 109	$ 114	$ 57	$ 94	$ 73	$ 603				
Approach 3	Week 1	Week 2	Week 3	Week 4	Week 5	Week 6	Week 7	Week 8		
Hours Utilized	126	105	0	78	82	133	137	67	728	
Waste Hours	27	18	133	32	51	0	16	46	277	38%
Total Cost	$ 2,790	$ 2,340	$ 2,530	$ 2,070	$ 2,430	$ 2,430	$ 2,790	$ 1,061	1051	
Waste Cost	$ 492	$ 342	$ 2,530	$ 602	$ 932	$ -	$ 292	$ 729		
Avg hourly	$ 18.24	$ 19.02	$ 19.02	$ 18.82	$ 18.27	$ 18.27	$ 18.24	$ 15.84		
Tot hours	153	123	133	110	133	133	153	113		

Table 8 Weekly Production and Waste Cost

	3 approaches			total	waste	prod	waste cost
1	*raw hours*			$ 15,338.57	$ -	$ 15,338.57	0%
2	*best fit*			$ 15,777.07	$ 447.80	$ 15,329.27	3%
3	*admin priorities*			$ 20,591.33	$ 5,190.54	$ 15,400.78	25%
	$ 13,548.57	**$ 13,987.07**	**$ 18,441.33**		**$ 13,548.57**	**$ 13,987.07**	**$ 18,441.33**
$ 13,548.57	$ -	$ (438.50)	$(4,892.76)	**$ 13,548.57**	100%	97%	73%
$ 13,987.07	$ 438.50	$ -	$(4,454.26)	**$ 13,987.07**	103%	100%	76%
$ 18,441.33	$ 4,892.76	$ 4,454.26	$ -	**$ 18,441.33**	136%	132%	100%

Table 9 Actual Cost for Each Model