Managing a Data Analysis Production Line:
An Example from the Whitehead/MIT Center for Genomic Research

By
Scott A. Rosenberg

B.S.E. Computer Science, Princeton University (1995)

Submitted to the Sloan School of Management and the Department of Electrical Engineering/Computer Science in Partial Fulfillment of the Requirements for the Degrees of

Master of Business Administration
Master of Science in Electrical Engineering/Computer Science

In Conjunction with the Leaders for Manufacturing Program at the Massachusetts Institute of Technology
June 2003

Signature of Author ______________
Sloan School of Management
Department of Electrical Engineering/Computer Science
May 2003

Certified by ______________
Stephen C. Graves, Thesis Supervisor
Professor of Management

Certified by ______________
Roy E. Welsch, Thesis Supervisor
Professor of Management

Accepted by ______________
Margaret Andrews, Executive Director of Masters Program
Sloan School of Management

Accepted by ______________
Arthur C. Smith, Chairman, Graduate Committee
Department of Electrical Engineering/Computer Science

The author wishes to acknowledge the
Leaders for Manufacturing Program
for its support of this work.

He would also like to thank Chad Nusbaum,
Rob Nicol, Mike FitzGerald, Nabil Hafez, and
the team of finishers at the Whitehead Institute.

Finally, he would like to thank his wife, Carolee,
for supporting him while he had fun
in graduate school.

Managing a Data Analysis Production Line:
An Example from the Whitehead/MIT Center for Genomic Research

By
Scott A. Rosenberg

Submitted to the Sloan School of Management and Department of Electrical Engineering/Computer Science on May 1, 2003 in partial fulfillment of the Requirements for the Degrees of Master of Business Administration and Master of Science in Electrical Engineering/Computer Science

Abstract

Finishing is the final phase of the gene sequencing process employed in the Human Genome Project. During the finishing process, analysts inspect and resolve ambiguous regions of DNA. While other phases of gene sequencing have evolved into high-volume, automated processes, finishing has remained manual, time-consuming, and costly. This thesis studies finishing as an example of a broader class of processes it terms *data analysis production lines*. Unlike traditional production lines, data analysis processes typically consist of complex, information-oriented tasks in which skilled analysts play a central role. Because task complexity and analyst skill level tend to vary, however, processes like finishing are often difficult to monitor, predict, and control.

This thesis identifies a series of workflow policies and organizational changes designed to help managers regain control of data analysis production lines. Using insights gained from a six-month internship at the Whitehead/MIT Center for Genomic Research, it shows that process variability can be greatly reduced through disciplined workflow policies. Specific policy recommendations include: breaking complex tasks into simpler ones before assignment; limiting analysts' workloads; requiring analysts to process work in first-in-first-out order; and, matching task complexity to analyst skill level. A simple modeling framework serves as the basis for proving the efficacy of these policies. The thesis concludes with an analysis of the organizational dynamics at Whitehead. A new performance review and incentive system is proposed as a means of encouraging better teamwork and knowledge sharing between analysts.

Thesis Supervisor: Stephen C. Graves
Title: Professor of Management

Thesis Supervisor: Roy E. Welsch
Title: Professor of Management

Table of Contents

Introduction

In genomics, software development, and a variety of other emerging data-intensive industries, *manufacturer* seems a poor label for most companies. The products created by these companies are not physical; more often they are collections of analyses assembled by human experts who study, manipulate, and characterize large datasets. In the gene sequencing case considered by this thesis, expert analysts called *finishers* attempt to resolve ambiguities in DNA sequence through visual inspection and the application of various laboratory procedures. In *finishing*, as this process is called, and in other data-intensive businesses, experts play a central role in what this thesis terms a *data analysis production line*.

Data analysis production lines resemble traditional production processes in spirit but differ in some critical ways. Certainly, success in either type of process hinges on a detailed analysis of classic manufacturing parameters such as cycle time, yield, and quality. Automation is also an important, common lever for reducing system variability and cost. In many ways, though, similarities between the two types of processes end with these shared conceptual goals. Data analysis tasks tend to be lengthy, complex procedures that preclude simple automation, leaving human analysts to perform a majority of the truly valuable analytic work. These analysts, in turn, exercise more discretion over their tasks than typical production personnel, deciding which procedures to apply, what work to prioritize, and when to seek assistance from fellow co-workers. Thus, data analysis workflows differ from traditional production lines in important ways: tasks tend to be complex and human analysts play a central role in the production process.

These same characteristics that set data analysis processes apart also make them highly variable. Data analysis tasks are complex, but their true complexity is also difficult to predict a priori. Often, an analyst must begin a task before he or she can understand the time and resources needed to complete it. Similarly, analysts' skills tend to vary significantly. These differences are amplified when there is limited collaboration between analysts. Together, variations in task complexity and analyst skill make for a volatile process. At the Whitehead/MIT Center for Genomic Research, where this research was conducted, week-to-week output of the Finishing Group varied by as much as 50% of the mean. In short, the very complexity that necessitates expert analysis also renders it highly unpredictable.

Not surprisingly, highly variable processes like finishing are difficult to monitor, predict, and control. When output fluctuates as significantly as it has at Whitehead, managers find it difficult to identify and triage individual sources of variability amidst the overall system noise. Similarly, long-term productivity trends become obscured by short-term fluctuations. Planning in this environment quickly breaks down. Management finds it difficult, if not impossible, to forecast and plan future productive capacity. High process variability also takes a toll on a group's psychology. Analysts and management alike grow frustrated with their inability to predict productivity. Short-term initiatives tend to rule the day. Thus, high variability poses a variety of process control and morale problems. By most measures, data analysis processes are *out of control.*

Drawing upon insights gained during a six-month internship at Whitehead, this thesis studies data analysis production lines from two perspectives. First, it argues that the variability present in data analysis processes is often unnecessarily amplified by ineffective workflow policies. Disciplined approaches to how tasks are assigned to and managed by analysts can greatly reduce variability over that seen in Whitehead's current process. Second, this thesis argues that analyst performance variability can be reduced through teamwork and collaboration. Analysts' skills diverge not just as a result of differences in natural aptitude, but also from their having limited training and knowledge sharing opportunities. A proposal is put forth that encourages teamwork through changes to Whitehead's current performance review and incentive system.

In summary, this thesis targets the process control problems inherent in an emerging class of production processes whose primary output is data analysis. These processes are characterized by high task complexity and a heavy dependence on human experts – elements that combine to produce variable, difficult-to-control processes. This thesis proposes a number of policies designed to stamp out variability stemming from ineffective workflow practices. At the same time, it advocates a number of organizational changes that can lead to better collaboration and teamwork. These proposals are put forth in the context of work conducted at Whitehead, but have broad applicability to other data analysis production lines.

The thesis proceeds as follows:

Chapter 2, Genomics Background provides a background on modern gene sequencing technology. It concludes with a detailed discussion of the finishing phase of the gene sequencing process, providing insights into the characteristically iterative and variable nature of finishing.

Chapter 3, Formalizing the Problem builds upon the discussion of Chapter 2, providing historical data on the variability of the finishing process and the problems it has created. It also establishes a modeling framework that forms the basis for the analyses of Chapters 4 through 7.

Chapter 4, Task Bundling shows that Whitehead's practice of assigning complex projects to its analysts unnecessarily amplifies the variability of their output. Instead, a case is made for breaking complex projects down into simpler, single-task projects.

Chapter 5, Task Assignment shows that proper timing of new task assignments ensures efficient analyst utilization while keeping work-in-process levels low. The chapter concludes by providing guidance on how to optimally assign tasks to a group of analysts with disparate skill levels.

Chapter 6, Workflow Management shows the unnecessary variability and productivity losses that can occur when analysts are given large workloads and the discretion to prioritize tasks as they see fit. Stricter control over this aspect of the workflow removes a potent source of process variability.

Chapter 7, Teamwork outlines several team-based scenarios that can boost productivity, lower process variability, and improve organizational communication and knowledge sharing.

Chapter 8, Organizational Studies examines the structural, political, and cultural aspects of Whitehead's organization that inhibit higher performance. It concludes with a recommendation for a new performance review and incentive system that emphasizes teamwork and training.

Chapter 9, Conclusion ties the discussion of the preceding chapters together and offers some perspectives on how their conclusions can be applied outside the field of genomics.

2 Genomics Background

This chapter provides a brief introduction to the Human Genome Project and today's gene sequencing process. It concludes with a detailed discussion of the finishing portion of the process, which is the chief focus of this thesis. The objective of this chapter is to provide the reader with a sufficient background in genomics to understand finishing process optimization.

2.1 The Human Genome Project

The Human Genome Project (HGP) began in 1990 with the simple but ambitious goal of sequencing the entire genetic makeup of the human species. While simpler organisms had previously been sequenced, the HGP represented a significant increase in both genomic size and complexity. Primary responsibility for the sequencing fell to large genome centers like the Whitehead Institute at MIT, Washington University, Baylor University, and the Sanger Center in Great Britain, with dozens of smaller centers around the world also contributing.

In 2000, a draft sequence of the human genome was published. For much of the research community, this represented a fulfillment of the major objectives of the HGP. However, the publishing of the draft sequence meant only that basic sequencing was complete. The draft sequence contained many absent, ambiguous, or conflicting regions of DNA. In the time since the draft's publication, genome centers like Whitehead have focused their energies on systematically clarifying these problematic regions. This clarification process is called *finishing*. In April 2003, timed to coincide with the fiftieth anniversary of the discovery of DNA by Watson and Crick, a final version of the human genome was at last completed.

In the thirteen years that comprised the HGP project, the technologies underlying gene sequencing were revolutionized in both technique and process speed. In 2000 alone, three times as much DNA sequence data was produced than in the first five years of the HGP. Indeed, the science behind gene sequencing has evolved so significantly that sequencing of the mouse (a similarly complex genome) is budgeted to cost less than 20% of the HGP. Looking forward, we can expect to see a continued acceleration of the gene sequencing process until, as some predict, we are capable of sequencing an individual's genetic code in just one day.

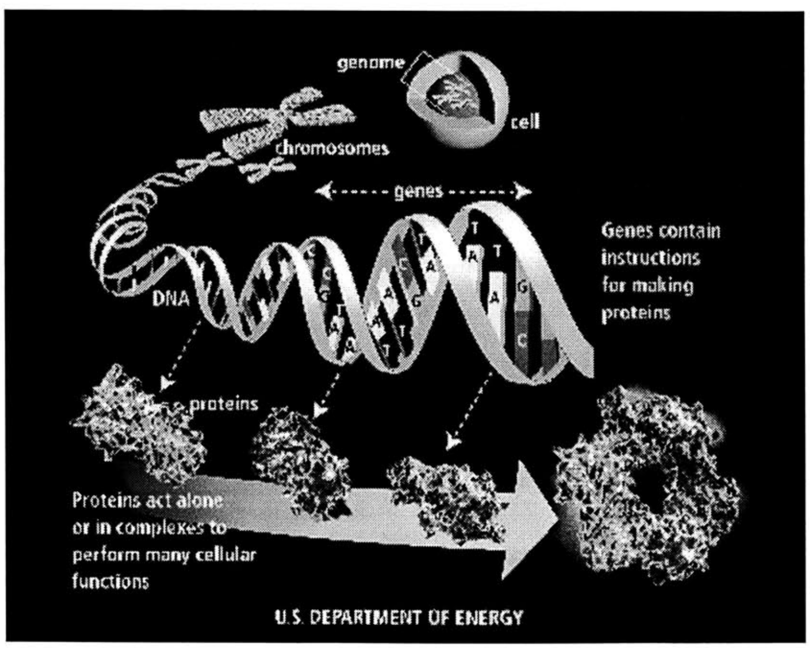
In the meantime, the HGP has laid down a lattice of genetic knowledge upon which decades of future medical and evolutionary research will be transacted. Already, scientists have begun to conduct broad cross-genomic comparisons between humans and other species to determine which genes have been preserved through evolution and which genes are uniquely human. From a medical perspective, too, doctors will begin to use HGP data to develop tests that determine whether a person is at risk for certain diseases. If a person is proven to be at risk, scientists may be able to introduce corrective genetic material into his or her body. Data from the HGP is a critical enabler for this type of medical research.

2.2 DNA Basics

Deoxyribonucleic acid (DNA) is the genetic building block upon which all known life regulates its daily function and long-term evolution. DNA is comprised of long strings of just four nucleotide bases called adenine (A), guanine (G), cytosine (C), and thymine (T). Active sequences of DNA that are hundreds or thousands of base pairs long, called *genes*, are translated into *proteins* during the course of cell activity. Proteins, in turn, regulate all of life's most basic functions, including: forming structural elements in cells; acting as catalysts for chemical reactions in the body; and regulating the generation and activity of still other proteins.

Structurally speaking, DNA is a stable polymer that arranges itself into a helical structure of the sort shown in Figure 1. Long sequences of nucleotide bases form one half of the structure. Each base also bonds to its complementary base in the other half of the structure: A pairs with T and G pairs with C. Thus, a sequence of "ATTGC" bonds to its complementary sequence "TAACG". Each additional pair of bases in the structure tends to rotate the molecule by several degrees. Over long sequences, a helical structure results, giving the DNA molecule its *double helix* appearance, as first recognized by James Watson and Francis Crick in 1953.

Figure 1. Relationship between cells, chromosomes, DNA, and proteins.[1]



[1] Source: U.S. Department of Energy, http://www.ornl.gov/TechResources/Human_Genome/publicat/primer2001/1.html

Every cell in the human body contains a complete copy of a person's entire genetic makeup. Forty-six *chromosomes*, each an average length of 130 million base pairs long, sit inside the nucleus of every cell. These 46 chromosomes represent 23 pairs of mostly identical chromosomes. One chromosome in each pair is derived from each parent. Slight differences between the genes on these parental chromosomes, and the manner in which they are manifested, are what cause people to demonstrate attributes to varying degrees from each parent.

Differences between individuals are a complex function of variations in both the genetics of those individuals and the environment in which they are raised. The genetic contribution to these differences derives from the tendency of genetic material to mutate and recombine over many generations. The first of these differentiation processes, mutation, is rare but influential. On average, human beings differ genetically by no more than 1 in 1000 DNA base pairs. Mutation is the source of this seemingly miniscule 0.1% difference. The second differentiation process, recombination, plays a more significant role in human diversity. Recombination occurs when material from one chromosome in a pair is exchanged with similar material in the other chromosome. Mutation and recombination, together with environmental pressures that favor some genes and gene combinations over others, are responsible for the incredible genetic diversity observable in all living things.

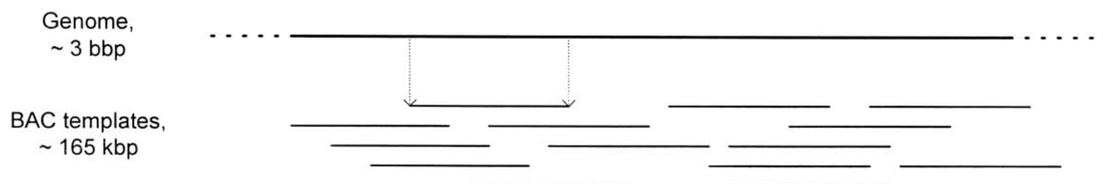
All told, the human genome consists of more than three billion DNA base pairs and an estimated 30,000 genes. For many years following DNA's discovery by Watson and Crick, scientists could sequence only a few DNA base pairs at a time. In the 1980's, it became possible to sequence hundreds and thousands of DNA base pairs. Finally, in the 1990's, genome centers like Whitehead gambled that they could apply recent advances in DNA science to sequencing the entire human genome. This ambitious, but ultimately successful, gamble turned into the gene sequencing process described in the next section.

2.3 Gene Sequencing Basics

Today's state-of-the-art gene sequencing technology proceeds by breaking large DNA samples into small segments, determining the exact DNA sequence of those small segments, then reconstructing sequence from these segments into a composite view of the original sample. The description that follows is time-sensitive because the technology it describes is constantly evolving and varies to some degree between genome centers. Nonetheless, this divide-and-conquer process forms the basis for all modern, high-volume sequencing operations.[2]

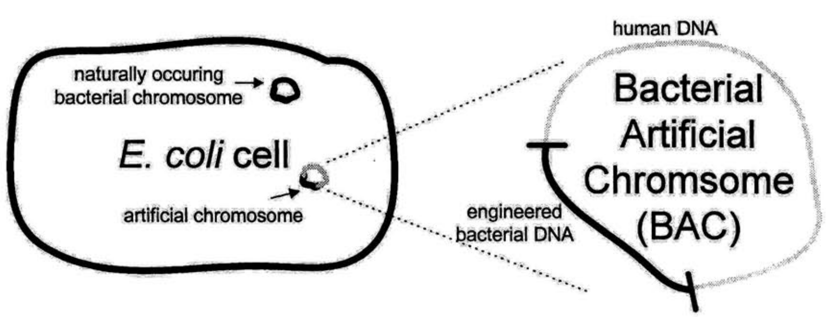
The process begins with the identification of the individual specimens to be sequenced. In the case of the HGP, DNA was donated by a small set of consenting, anonymous individuals. After the DNA is purified, enzymes are used to break it down into smaller segments. From the mix that results, segments with a length of approximately 165,000 base pairs (165kbp) are isolated. These samples, called *BAC templates*, constitute a library of genetic material. In the HGP, for example, templates derived from one individual became known as "RP-11." Figure 2 illustrates the process of breaking down a genome into templates.

Figure 2. Splitting of genome into 165kbp templates.



A sample size of 165kbp is chosen such that an engineered version of the bacteria *Escherichia coli*, more commonly called *E. coli*, can be tricked into carrying and reproducing the human genetic material. By wrapping the 165kbp template with engineered bacterial DNA, scientists fool the bacteria into assuming the human DNA is a circular chromosome just like the one already present in *E. coli*. When the bacteria replicate, instructions in the DNA wrapper tell them to also duplicate their human DNA. The combination of a human genetic template and engineered bacterial DNA is called a Bacterial Artificial Chromosome, or *BAC*. Figure 3 illustrates the basic relationship between a BAC and its host *E. coli*.

Figure 3. BAC structure and E. coli host.



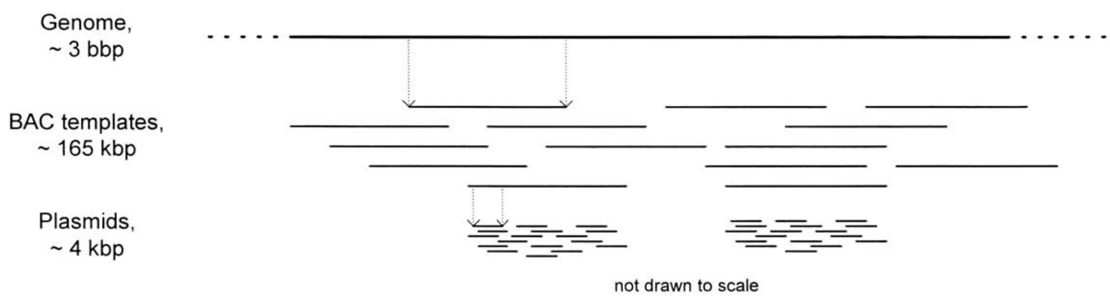
In this sophisticated manner, the bacteria's machinery is co-opted to replicate human genetic material many millions of times in just hours, the same amount of time it takes a single

[2] U.S. Genomics was recently funded on the promise that it could deliver a technology for directly "reading" long sequences of DNA.

bacterial cell to reproduce itself millions of times under ideal conditions. Mass-replication of the human genetic sample is critical to the gene sequencing process. It would be difficult to manipulate and sequence a single strand of DNA. By amplifying the original human genetic material, *E. coli* help produce volumes sufficient for the many reactions that comprise the gene sequencing process.

The BAC is still a large unit of genetic material that cannot be directly sequenced. Therefore, it must be further broken down before its DNA can be deciphered. To accomplish this further reduction in sample size, BACs are sheared through a physical process. The resulting mix is then filtered for DNA segments of uniform size, usually between 4kbp and 10kbp. Once isolated, these segments are tagged with bacterial DNA, becoming what is known as a *plasmid*. Plasmids are then inserted into host *E. coli* for amplification, in much the same manner that BACs are amplified. Figure 4 illustrates the successive processes of first breaking a genome into 165kb templates and then into 4kb plasmids.

Figure 4. Two phase break-down of genome into BACs and then plasmids.

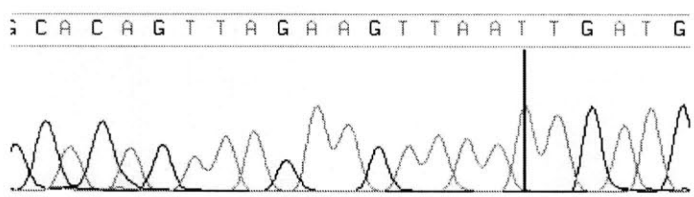


Plasmids are the units upon which direct DNA base pair *detection* is conducted. After being amplified through *E. coli*, plasmids are stripped from their host bacteria. The resulting material is then placed in a solution containing special DNA base pairs that are tagged with a fluorescent dye. By raising the temperature of the solution, the plasmid DNA, which normally resides in a paired helical structure, can be induced to separate. When the temperature is lowered, an enzyme in the solution reconstructs the helical structure by grabbing base pairs from the surrounding solution. Whenever the enzyme selects a dyed base pair, however, the reconstruction process stops, leaving a DNA segment that is prematurely terminated by a dyed A, T, G, or C. By cycling the heat many times, technicians can produce a solution containing a wide array of segment sizes. Figure 5 illustrates the resulting, dyed sub-segments:

Figure 5. Plasmid dye-tagging process. [3]

The resulting solution of dye-terminated segments becomes the input to *detection*, the last of the laboratory stages of the gene sequencing process. The solution of dyed plasmid segments is placed at one end of a long capillary. A charge causes the DNA to migrate through this capillary, with smaller segments racing ahead of larger segments because of their lighter molecular weight. At the end of the capillary, where the segments gradually emerge, a laser illuminates the dyed base pairs at the end of the DNA molecules. A sensor detects the continuously varying illumination and records it in a data file. A piece of software then analyzes this data and makes a base-pair determination. Figure 6 shows the illumination seen by the sensor and the base-pair determination.

Figure 6. Example output from detection process.



What remains of the gene sequencing process is strictly information processing. A software tool called an *assembler* attempts to match plasmids from a BAC by similarities in their sequence. If everything works correctly, the assembler will reconstruct a single view of a BAC's underlying sequence. Figure 7 illustrates this reconstruction process. The top rows represent the sequence data for eight plasmids. Note that the assembler has aligned each plasmid such that their base pairs at each column match exactly. The last row represents the assembler's reconstructed, *consensus* view of the underlying BAC sequence.

[3] This graphic is from an animated educational toolkit provided by the National Human Genome Research Institute, http://www.genome.gov. The toolkit is available at http://www.genome.gov/Pages/Education/Kit/main.cfm

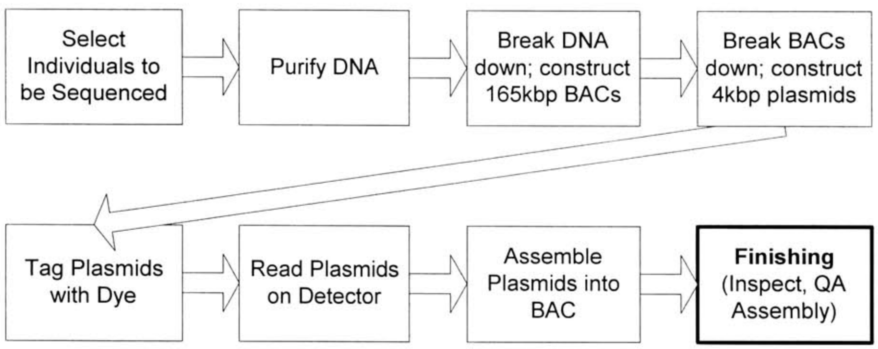
Figure 7. Example assembly with eight reads and consensus.

```
sequence data        CTTTCTCACAGAAGCTCCCTCTAGTTTT
from eight           CTTTCTCACAGAAGCTCCCTCTAGTTTTCTATG
plasmids             CTTTCTCACAGAAGCTCCCTCTAGTTTTCTATGTCTGAAAGTTCTTGAGGAT
                     CTTTCTCACAGAAGCTCCCTCTAGTTTTCTATGTCTGAAAGTTCTTGAGGAT
                     CTTTCTCACAGAAGCTCCCTCTAGTTTTCTATGTCTGAAAGTTCTTGAGGAT
                                             TAGTTTTCTATGTCTGAAAGTTCTTGAGGAT
                                                       TGTCTGAAAGTTCTTGAGGAT
                                                               AGTTCTTGAGGAT
consensus sequence as
determined by assembler
                     CTTTCTCACAGAAGCTCCCTCTAGTTTTCTATGTCTGAAAGTTCTTGAGGAT
```

In many cases, unfortunately, the assembler fails to construct a complete rendition of the BAC due to insufficient or poor quality sequence data. The reasons for this failure and how it is managed are the focus of the next section.

Figure 8 recaps the overall gene sequencing process flow.

Figure 8. Gene sequencing process flow for Human Genome Project.



2.4 The Finishing Stage of Gene Sequencing

Finishing represents the stage of gene sequencing process where analysts triage problematic BAC assemblies. Wherever the assembler cannot find DNA sequence for a region in a BAC assembly, a *gap* results. Human analysts, called *finishers*, attempt to close these gaps through a variety of laboratory techniques. Generally speaking, finishers triage two types of gaps. A *captured gap* is spanned by genetic material from a single plasmid. The finisher may be able to perform a lab procedure on the plasmid in order to discover the missing sequence. An *uncaptured gap*, on the other hand, is not spanned by a plasmid.[4] The finisher must use other, more complicated techniques to discover the missing DNA sequence. For a variety of reasons, uncaptured gaps usually prove more difficult than captured gaps. Finishers perform other functions as well, including clarification of ambiguous DNA sequence. However, gap closure is a finisher's primary function and is therefore the chief focus of this thesis.

[4] A gap occurs between the sequences of two or more plasmids. Thus, the assembler has no basis for joining the sequence on either side of the gap.

A typical finishing group, like the one at Whitehead, consists of as many as twenty to forty finishers. Finishers employ a variety of software tools that help them track and analyze gaps. In some cases, finishers can close a gap by simply editing the data already present in the BAC assembly. In other cases, finishers must order laboratory procedures in order to discover missing sequence information. In classic manufacturing terms, the finishing process represents the inspection, quality assurance, and rework phases of the gene sequencing process. The high-volume portion of the gene sequencing process, called *production sequencing*, generates most of the sequence information in a genome without quality problems. Where problems arise, however, it is the responsibility of finishing to correct this sequence.

Production sequencing can yield low-quality data for a variety of reasons: the DNA in a plasmid may prove toxic to *E. coli*; the DNA may resist certain chemicals in the laboratory; or, an operator error may have occurred somewhere in the process. There are many other reasons for why the process may fail, few of which are fully understood. In general, the real reason for failure is difficult to precisely determine. Different failure modes often manifest similar symptoms. Moreover, multiple problems may occur concurrently. Finally, even if it were technically feasible to determine why a failure occurred, it is often not economical to do so.

While it is often difficult to determine why a region of DNA did not sequence correctly, having such information would be extremely valuable. Knowledge about the sequence in and around a gap can help the finisher to select the most appropriate laboratory technique. For example, one laboratory technique employed by finishers, called *resequencing,* repeats the basic process used in production sequencing but utilizes more sophisticated chemistries. While resequencing often succeeds where basic sequencing did not, both technologies fail when applied to DNA sequence containing long stretches of G and C nucleotide bases. Thus, if the original gap occurred because of GC-rich content, the finisher would not be able to determine whether resequencing is likely to succeed. In essence, finishers find themselves in a catch-22: to select an appropriate laboratory procedure, they must understand the underlying sequence, but the sequence is missing.

The scenario just outlined is one of the many sources of uncertainty a finisher faces. In their decision making process, finishers must make educated guesses about the underlying sequence and the likelihood that various laboratory techniques will succeed. Their decision is influenced by the condition of the DNA near the gap. It is also influenced by the ability of their informatics tools to highlight those conditions. Most importantly, finishers' decisions are guided by their skill and experience. Whereas some experienced finishers may be able to close a gap based on the information already present in an assembly, less experienced finishers may feel they need laboratory work. Worse still, some finishers may be unable to close the gap without help. In short, finishing is fraught with uncertainty that is amplified by differences in finisher skill.

Given the uncertainty inherent in the finishing decision process, finishers are often unable to close a gap and must try again until they succeed in closing the gap. The gap is closed when all base pairs in a BAC are accounted for and of sufficiently high quality.[5] Each attempt generates information that offers new insights into how the finisher should proceed. For example, the gap may have been partially closed, indicating that the previous procedure worked, albeit incrementally. Alternatively, a failure may indicate that the underlying DNA is resistant to the chosen procedure. In still other cases, the procedures may fail uniformly, raising the possibility that the lab committed an error. With the information they gain at each attempt, finishers proceed in a trial-and-error, iterative fashion until they succeed in closing the gap.

Several other finishing process characteristics are important to note because they play an important role in the finishing workflow. First, BACs are spliced out of a genome by means of an enzymatic process that is unaffected by the sequence problems that may lead to a gap. As a result, a single BAC assembly may contain one or more gaps. This "gap bundling" phenomenon adds another layer of uncertainty to the process and is the subject of Chapter 4. Second, because finishing is an iterative, complex workflow, finishers often develop a familiarity with specific gaps. Because their knowledge of a gap may be subtle and intuitive, it is often not written down. As a result, it is difficult to transfer work-in-process gaps between finishers. At Whitehead, finishers have historically retained full responsibility for a gap until it is closed. The inability to transfer gaps between finishers hampers collaboration and heightens process variability. These issues are the subject of discussions in Chapters 7 and 8.

2.5 Summary

This chapter explained the divide-and-conquer strategy that characterizes today's gene sequencing process. Specific attention was given to the finishing phase of the process, which readers from a manufacturing background will recognize as the inspection, quality assurance and rework phases of a production process. No doubt, these same readers observed with some alarm that finishing is an iterative, trial-and-error workflow. Uncertainty, like that seen in finishing, is usually the harbinger of an out-of-control process. In the next chapter, we see the extent to which this is the case and identify a framework for studying workflows like finishing.

[5] NHGRI defined standard quality metrics for all genome centers. These limited, for example, the number of consecutive base pairs below a specific certainty level. Certainty, in turn, was estimated by means a software routine shared across all genome centers.

3 Formalizing the Problem

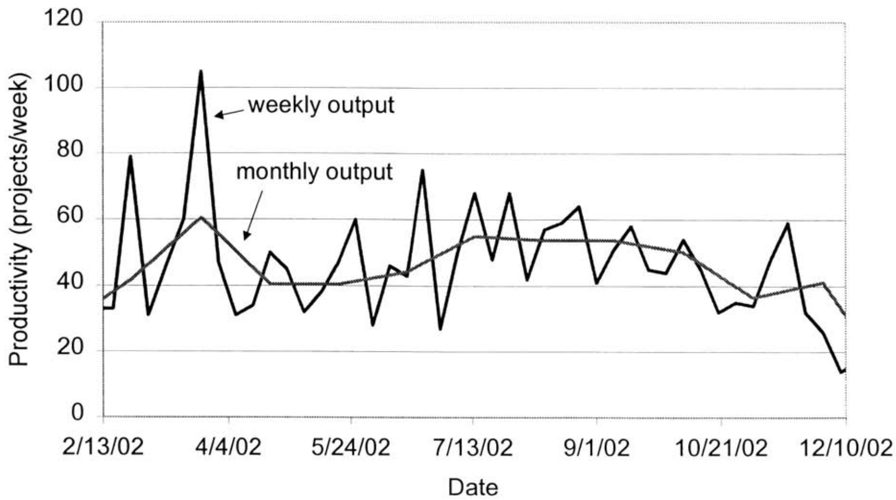
The description of finishing in the previous chapter should have given the reader a sense for the iterative, variable nature of the process. This chapter begins by presenting data showing the historical variability of Whitehead's finishing process. It then examines why this variability is problematic from a process control and planning perspective. With this discussion as the backdrop, the chapter poses the central problem statement of this thesis: How can one mitigate process variability in data analysis workflows like finishing? The chapter concludes by presenting a modeling framework that serves as the basis for analyzing solutions proposed in the remainder of the thesis.

3.1 Finishing Process Variability

Figure 9 shows the output of Whitehead's Finishing Group in 2002. In this graph, output is measured as the number of projects completed by the group over the period of a week or a month. A project is defined as the set of all finishing work needed to clarify the DNA sequence in a single BAC assembly. As subsequent discussion will show, project complexity can vary significantly, making project completions an imperfect measure of productivity. Nonetheless, it is the measure currently employed by Whitehead and therefore a sensible point from which to start the discussion.

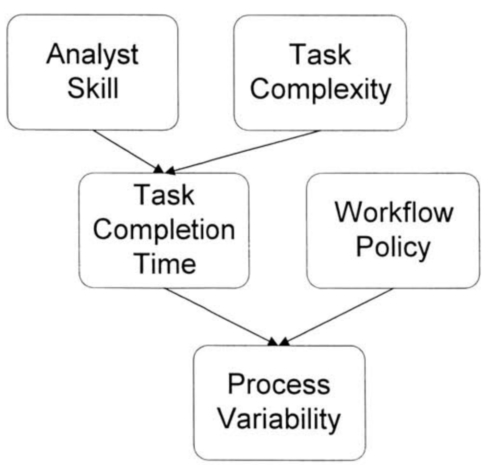
As one can see from the graph, the output of Whitehead's finishing process is highly variable. The standard deviation of output relative to its mean (σ/μ) in 2002 is 36% and 20% on weekly and monthly bases, respectively. In one two-week period in June, for example, output dropped from 75 projects in one week to 27 projects in the next week.

Figure 9. Whitehead Finishing Group output in 2002. [6]



Variability in data analysis processes like finishing stems from a wide variety of sources. However, three systemic sources of variability stand out: high *task complexity*; variations in *analyst skill level*; and, ineffective *workflow policies*. Figure 10 provides a conceptual framework for thinking about how these factors contribute to process variability. Variations in analyst skill and task complexity make it difficult to predict how long an analyst will need to complete a task, independent of the workflow policies controlling how tasks are assigned and managed. When workflow policies are ineffective, they serve merely to amplify the uncertainty present in task completion time. Together, these three sources of uncertainty create a highly variable process.

Figure 10. Key sources of variability in data analysis processes.

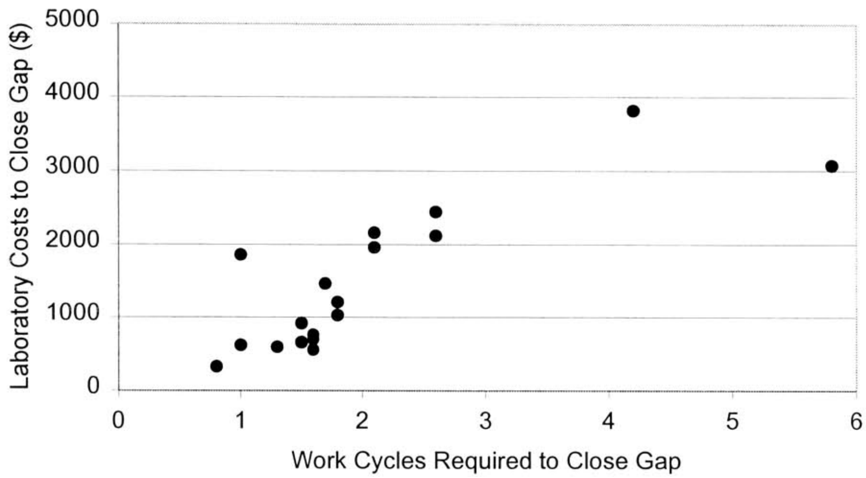


Analysts' skills vary according to differences in natural aptitude, experience, and training. Figure 11 illustrates just how large these discrepancies can be. It shows the average time and

[6] Monthly completion rates are converted to weekly rates (i.e. divided by ~4) in order to facilitate comparison.

laboratory costs needed by finishers at Whitehead to close a captured gap. Each dot in the graph represents the performance of a single finisher. A finisher's performance is calculated as the average time and cost across all of the captured gaps he or she completed in 2002. Time, in this case, is measured as the number of work cycles (trial-and-error attempts) a finisher needed to close a gap.[7] Cost is estimated as the sum of materials cost and an allocation of laboratory overhead.[8] As one can see from the graph, the performance discrepancy between finishers is large. Some analysts require three times the resources to complete the same task. For a manager, this means that task completion time is difficult to predict because it depends intimately on which analyst receives the task.

Figure 11. Average per-gap time and cost for finishers to close captured gaps.

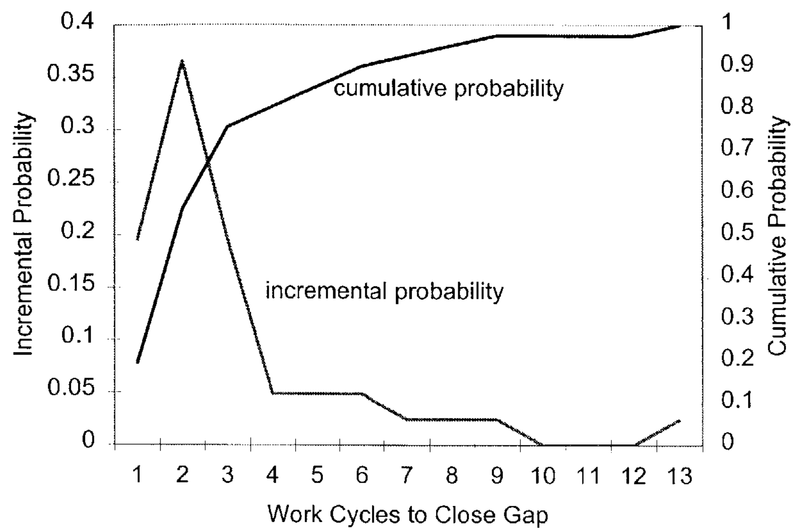


Wide variations in task complexity also make it difficult to predict task completion time, even for a single analyst. Finishers often find that two seemingly similar tasks require significantly different levels of effort. Figure 12 shows the distribution of work cycles needed by one of Whitehead's more senior finishers to close captured gaps. The incremental probability line shows the percentage of captured gaps closed by the finisher in any given work cycle. The cumulative probability line shows the percentage of all gaps closed by the given work cycle. While this experienced finisher completes over 50% of his captured gaps in just one or two cycles, 10% of his gaps required more than 6 cycles. Clearly, even at the hand of a highly skilled finisher, accurately predicting task completion time can be difficult.

---

[7] A finisher may spend different amounts of time in each work cycle making this is an imperfect measure. This issue is discussed shortly.
[8] Robert Barrett, a project manager at Whitehead, served as a source for these rough cost estimates

Figure 12. Observed time to close captured gaps for one expert finisher.



The third basic source of process variability is workflow policy. In this thesis, the term *workflow policy* refers to all operational decisions that affect the movement of work through the process. In some cases, workflow policy is the product of explicit managerial decisions. For example, a manager may tell his or her analysts to prioritize certain work over others. Workflow policy may also emerge in a distributed, ad hoc fashion. For example, an individual analyst might segment his or her day, performing some types of tasks in the morning but others in the afternoon. Each of these policy decisions has a potential impact on process variability. If, as in Whitehead's case, analysts are lent wide discretion over how they manage their workload, process control problems can be unnecessarily amplified.

This thesis analyzes opportunities to remove uncertainty from data analysis workflows like finishing. Workflow policy is the initial focus. In Chapters 4 through 7, newpolicies regarding task assignment and management are identified that substantially reduce process variability over Whitehead's current practices. Then, in Chapters 7 and 8, we turn to analyst skill, focusing on teamwork and incentives as a means to normalize analyst performance.

3.2 *The Evils of Variability*

Variability in the finishing process has numerous deleterious effects, including:

Large work-in-process (WIP) queues. To buffer themselves against the low utilization that often results from process variability, finishers must accumulate large WIP queues. Large queues, in turn, lead to longer project cycle times. They also tempt finishers to prioritize work according to short-term, often sub-optimal objectives (e.g. the desire to boost output before a performance review). When finishers *game* their workload like this, they merely amplify process variability.

Reduced process flexibility. Long cycle times reduce management's flexibility to plan future production. At Whitehead, difficult HGP projects often took a year or more to complete. Long cycle times act as a liability against future productive capacity. Resources must be committed far into the future, precluding their use on new projects that may emerge.

Capacity planning difficulties. High process variability makes it difficult for management to monitor and plan productive capacity. In Whitehead's case, process variability in the HGP was so significant that it took months to identify a long-term trend towards reduced output. In the meantime, preventative action like better training and additional hiring was not taken, causing the group to nearly miss its production deadlines.

Process control difficulties. Even when management is cognizant of its process variability problems, identifying and triaging individual sources of that variability may be difficult in systems as noisy as the finishing process. At Whitehead, variability seemed to emanate from all corners of the system at once: lab yields, finisher performance, task complexity, and workflow policy. With so many sources of variability, it may be difficult to know where to start.

Negative psychological impacts. Faced with high variability, limited insight into its causes, and an aggressive deadline, a manager may react with short-term policies that are ineffective or counterproductive. At Whitehead, efforts to boost short-term output often had the effect of reducing long-term productivity. Process variability can also have deleterious effects on line workers. Finishers grew frustrated with their own unpredictable output and reacted negatively to management's short-term initiatives.

3.3 Formal problem statement

Finishing appears to be a uniquely iterative, analytical workflow prone to high uncertainty. In many ways, though, it resembles some familiar processes. In a technical support call center, for example, operators field phone calls from customers facing a variety of problems. In some cases, the problem can be resolved immediately; in other cases, it requires multiple phone calls or escalation to a supervisor. During the course of the phone call, the operator may walk the customer through several procedures before finding a successful resolution. The call center's problems mirror those facing finishing: tasks are complex and difficult to predict a priori; they may require multiple passes; and, success pivots on the operator's communication and deduction skills.

In considering solutions to the Finishing's variability problem, we are challenged to think about the broader class of processes that it represents. Because they are analytical, information-intensive production processes, this thesis refers to finishing, technical call centers, and similar processes as *data analysis production lines*. Data analysis production lines generally share the following characteristics:

The product is information analysis. Though physical material may be manipulated during the process, the output of the production process is informational. In some cases, like finishing, the product is a quality-assurance seal. In other cases, like the technical call center, the product is advice.

Human analysts perform a majority of the value-added tasks. Unlike many production operations, data analysis tasks are usually complex enough to preclude simple automation. Human analysts fill the void, providing the necessary breadth and dexterity.

Analyst productivity varies significantly. An unfortunate byproduct ofa heavy reliance on human capital is that their skill levels tend to vary significantly. Analysts bring different levels of natural aptitude, experience, interest, and learning potential to their jobs. These differences inevitably manifest themselves in skill and performance discrepancies.

Quantifying analyst skill and task complexity is difficult. It isoften difficult to objectively measure analyst skill and task complexity because they are interdependent. Analysts will excel in different tasks. Yet management must find an equitable means to measure and incentivize analyst performance.

Definition of task completion may be subjective. In some cases, like the call center that offers technical advice, it may be difficult to determine when an analysis task is complete. The analyst or the supervisor must make a subjective determination to cease work on a task because it meets some quality metric or would not benefit from additional effort.

The workflow is iterative. Because task complexity and analyst skill level varies so significantly, analysts rarely know with certainty which decision is best. As a result, analysts must often proceed in a trial-and-error, iterative fashion until the task is completed successfully.

While this thesis focuses on specific data and examples gathered from the Whitehead's genome finishing process, its lessons are targeted at the broader class of data analysis production lines. In each, management must control the production line in spite of its high natural variability.

3.4 *Modeling data analysis workflows like finishing*

Before proceeding, we first establish a formal framework for thinking about data analysis production lines. This section defines the terminology, metrics, and modeling representations that form the basis for analyses in the remainder of the thesis.

3.4.1 Terminology

The following terms are used repeatedly throughout this document. Some of the terms have corollaries in other industries; some have a meaning that is particular to the finishing process

Analyst. This term refers to the human expert performing the bulk of the value-added analysis task in the production process. At Whitehead, the analysts in the finishing process are called ***finishers***. These terms are used interchangeably throughout this document.

Task. This term refers to a single, indivisible unit of work. In Whitehead's Finishing Group, the act of ***closing a gap*** is the most common task handled by a finisher.

Project. In Whitehead's Finishing Group, this term refers to the unit of work assigned to an analyst. A project may consist of one or more tasks. At Whitehead, the terms ***BAC*** and project are used interchangeably because projects are usually formed around BAC assemblies.

Work Cycle. This term refers to a single iteration of work performed by an analyst during his or her attempts to complete a project. At Whitehead, each work cycle usually entails ordering laboratory procedures. Because a project may contain more than one gap, finishers often work on several gaps in each work cycle.

Procedure. This term refers to an analytical or laboratory method employed by an analyst in the process of trying to complete a task. At Whitehead, the terms ***laboratory procedure*** and ***lab work*** are also used.

3.4.2 Key metrics

This thesis concerns itself primarily with techniques for reducing the variability inherent in data analysis production processes. This section describes how output and variability are measured.

Measuring output. Whitehead has historically used *projects* as its chief measure of productivity. Because a project is often comprised of multiple tasks, however, this measurement technique obscures the real level of work required by a project. For this reason, productivity in this thesis is usually presented in terms of both project and task completions rates.

Measuring variability. In the analyses and simulations presented, process variability is generally measured as the standard deviation of output divided by its mean over some time frame. Mathematically, this is σ/μ. Variability is measured on weekly or monthly bases.

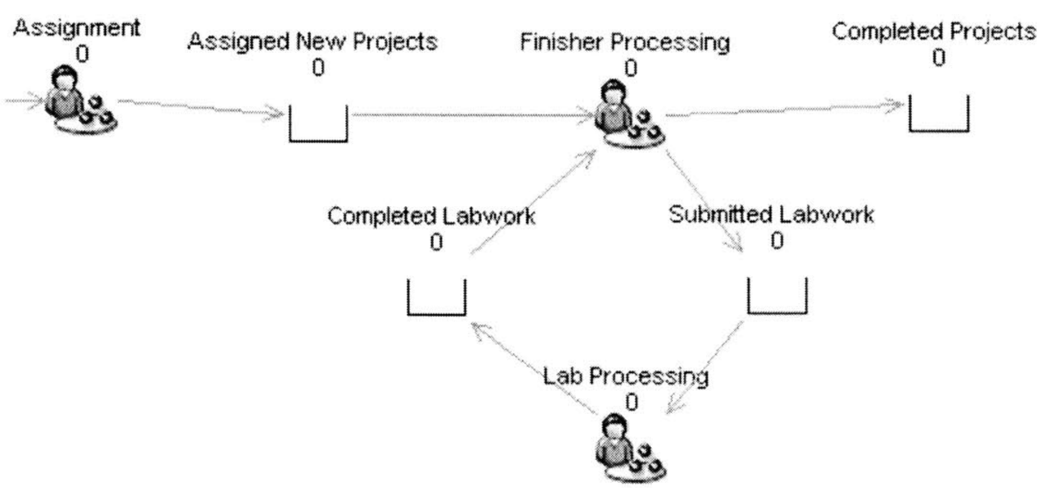
3.4.3 Modeling Considerations

Modeling complex systems inevitably pitches the desire for accurate representation against the need to be economical and focused in the model's design. Because this thesis focuses on variability stemming from task complexity, analyst skill, and workflow policy, these parametersare given priority in the modeling effort. Other system characteristics, like variability stemmingfrom variations in lab yields or delays, are given less attention. As a result, the models put forth in this thesis should be interpreted in a qualitative light. They are not perfectly representative of the dynamics at work in the finishing process.

3.4.4 Modeling the Finishing Workflow

A workflow simulation package called Simul8 is used to model the finishing workflow. Figure 13 illustrates how the workflow of an individual finisher is modeled in Simul8 for this thesis:

Figure 13. Model finishing workflow.



Newly assigned projects, generally assumed to be in infinite supply, are deposited intothe finisher's *Assigned New Projects* queue. After selecting a new project out of this queue, the finisher performs an initial cycle of work. When the finisher processes a project, he works on all unclosed gaps within that project. Gaps are considered independent; that is, success in one does not confer information about the others. Each captured gap is assumed to require one hour of work per cycle;

each uncaptured gap requires two hours of work per cycle. (The rationale for these parameters is discussed shortly.) The finisher works 35 hours per week.

If, after processing a project, the finisher has closed all of its gaps, he sends the project to his *Completed Projects* queue. If one or more of the gaps in a project remains open, the finisher sends the project to his *Submitted Labwork* queue for lab processing. Projects submitted to the lab are returned after an average of 70 hours (approximately two workweeks) with a standard deviation of 10 hours. The lab, while shared across all finishers, is assumed to have infinite capacity.

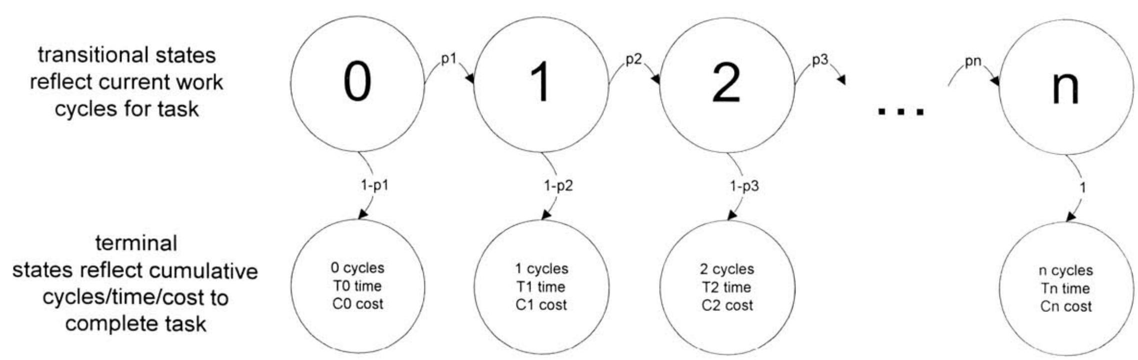
When the lab has finished processing a project, it returns the project to a finisher's *Completed Labwork* queue. The finisher may select projects out of either this queue or the *Assigned New Projects* queue. Finishers' policy for which queue they select from is one subject of study in this thesis.

3.4.5 Modeling Individual Tasks

The state of individual tasks (gaps) in the finishing workflow can be described as a Markov chain. The states in the chain represent the number of times the finisher has ordered laboratory procedures for the task. The transitions between states represent the probability that the task is either completed after a laboratory procedure or requires another round of work.

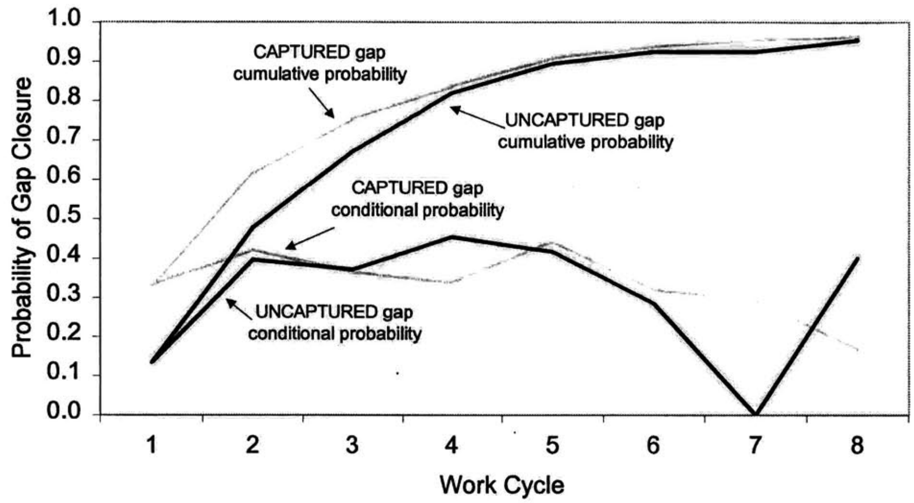
A Markov model allows us to capture two important characteristics of the finishing workflow: first, each additional cycle of work yields new information that affects a finisher's decision in the next cycle; and second, a finisher's chances of being successful changes according to this information. For example, a task may require one hour of work and complete with probability 0.5 in its first cycle; if the task is not completed, it might require an additional two hours of work and succeed with probability 0.3 in the second cycle. An example Markov chain is shown in Figure 14.

Figure 14. Markov model of finishing task state.



To represent the finishing workflow as a Markov chain, we must determine the state transition probabilities and the duration of time spent in each state. Considering transition probabilities first, Figure 15 shows the observed probability of gap closure by work cycle for Whitehead's finishers in 2002. Data for both captured and uncaptured gaps are shown. The *conditional probability* lines represent the chances that a gap is closed in its N^{th} work cycle given that it was not closed in the preceding N-1 cycles. The *cumulative probability* lines represent the chances that a gap is closed by the end of the N^{th} work cycle.[9]

Figure 15. Observed gap-closing probabilities at Whitehead.



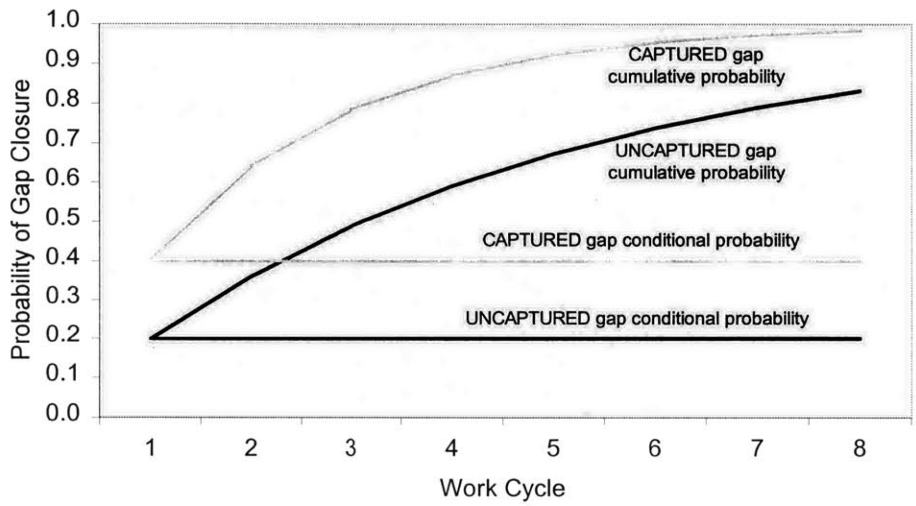
There are a number of biases in Figure 15 that bear mentioning. First, as will shortly be discussed, projects typically contain more than one gap. Because of the way projects are tracked at Whitehead, it is difficult to determine when individual gaps in a multi-gap project are closed. To generate the data above, a small population of 340 projects containing only one captured or uncaptured gap was pruned from the corpus of over 3000 projects completed by Whitehead. Because the sample data set was small, it is not perfectly representative of the overall population. There are also a number of data collection inaccuracies: gaps thought to exist were often later found to be data errors; uncaptured gaps were preferentially allocated to experienced finishers; and cycle counts were sometimes inflated by finishers that processed their labwork incrementally.

Because of these biases and a desire to simplify the analyses of this thesis, a simpler model of gap closing probabilities is used. For an average finisher, captured gaps are assumed to close with probability 0.4 at each and every cycle; similarly, uncaptured gaps are assumed to close with probability 0.2 at each cycle. The incremental and cumulative gap closing probabilities for captured

[9] Please note that there was limited data available for higher cycle counts. Thus, the dip in the conditional probability line for uncaptured gaps at the 7th cycle is a data collection anomaly.

and uncaptured gaps are shown in Figure 16. Though these simplified models exaggerate the differences between captured and uncaptured gaps, they capture the essential aspects of the original distribution: captured gaps close with higher probability on average in each cycle than uncaptured gaps; and, the probability of gap closure demonstrates an exponential approach to 1.0.

Figure 16. Idealized gap-closing probabilities.



Data on how much time finishers spend in each work cycle is more difficult to gather. Finishers do not currently log the time they spend on each gap. The time they spend also depends on the procedures they ordered. If a finisher opts to use one of the more sophisticated laboratory procedures, his or her processing time during that cycle will be higher. Using anecdotal evidence and a rough analysis of the number of gaps processed by a finisher each day, a simplified model of gap processing time is assumed in the modeling of this thesis. An average finisher is assumed to spend one hour per captured gap per work cycle and one and a half hours per uncaptured gap per cycle.

Table 1 summarizes the performance assumptions used in this thesis when analyzing individual finishers:

Table 1. Model of individual finisher performance.

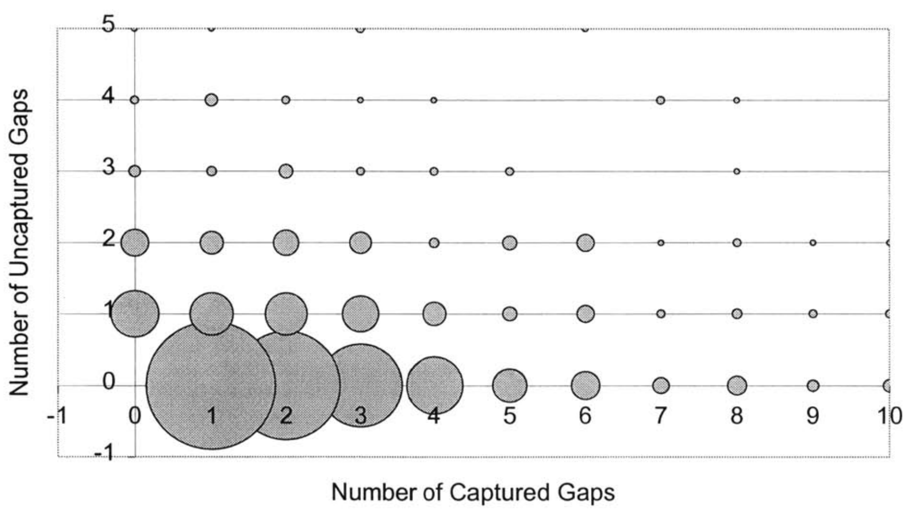
Conditional gap closing probabilities	
Captured gap / cycle	0.4
Uncaptured gap / cycle	0.2
Average time / gap / cycle	
Captured gap (hrs)	1.0
Uncaptured gap (hrs)	1.5

In summary, due to data collection and analysis issues, idealized models of the finishing workflow are assumed. Though these simplifications exclude certain features of the finishing process, they preserve its essential characteristics. They also greatly simplify the analyses and simulations of this thesis.

3.4.6 Task Distribution within Projects

Genomes are split into BACs by means of an enzymatic process, as described in Chapter 2. Because the splitting process is noisy, we might expect the incidence of gaps within BACs to be fairly random. Figure 17 shows the observed frequency of projects according to their gap count in Whitehead's portion of the human genome. Balloon size indicates the relative frequency of a project with [x,y] captured and uncaptured gaps. Projects with zero gaps ([0,0]) are excluded from the distribution because they generally require little finishing work.[10]

Figure 17. Observed gap distribution.



The mean rate of gap occurrence in Whitehead's projects was 2.1 captured gaps per project and 0.5 uncaptured gaps per project. Further analysis of the data revealed a 30% correlation between the incidences of captured and uncaptured gaps. Thus, gap distribution was not truly independent. The correlation arises out of the fact that a problematic region in the genome is likely to cause a number of captured and uncaptured gaps within a fairly localized region.

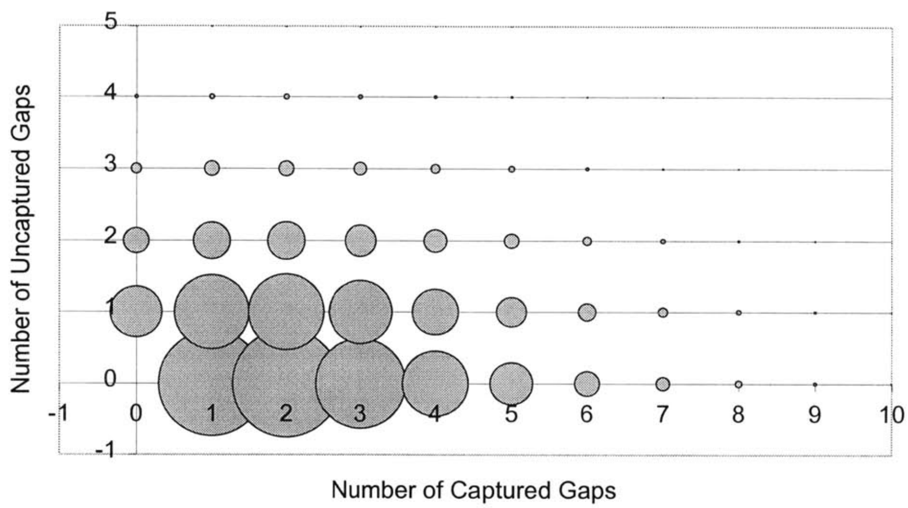
As in the previous section, there are a number of data collection problems that complicate direct use of the empirical data shown in Figure 17. Gap counts were often inflated due to problems in the BAC assembly; once these assemblies were resolved, gap counts were greatly reduced.

[10] Occasionally, however, there are low quality regions (i.e. not strictly gaps) that require significant work.

There was also an incentive for coordinators and finishers to inflate gap counts when they became the primary measure of productivity in 2002.

For these reasons, and to simplify our analyses, we make a number of assumptions about the rates at which gaps occur. First, we assume captured and uncaptured gaps occur independently. Second, we assume that the number of gaps in a project occur according to a Poisson process with the same observed means as the empirical data (i.e. 2.1 captured gaps per project and 0.5 uncaptured gaps per project). Figure 18 shows gap distribution according to this simplified model. Here again, projects with no gaps ([0,0]) are excluded from the distribution, as they require little or no finishing work.

Figure 18. Idealized gap distribution.



A number of discrepancies with the empirical model are easily spotted. Gap counts fall off more precipitously in the theoretical model. Also, projects with one uncaptured gap appear more common than in the empirical data. For the purposes of this thesis, however, these discrepancies are more than outweighed by the ease of analysis enabled by the theoretical model.

3.4.7 Modeling the Interaction Between Skill and Task Complexity.

The modeling discussion thus far has ignored the role that analyst skill level plays in the finishing process. Chapters 5, 7, and 8, however, analyze the role that skill plays in optimal task assignment, teamwork, and training. Thus, in addition to the *in-the-average* performance characteristics assumed above, we require a model of how finisher performance varies by skill. Three classes of finisher skill level are assumed: beginner, intermediate, and advanced. Assumptions about probabilities of gap closure and time per gap per cycle are shown in Table 2.

Table 2. Model finisher performance by skill level.

	Beginner	*Intermediate (average)*	*Advanced*
Conditional gap closing probabilities			
Captured gap / cycle	0.32	0.40	0.48
Uncaptured gap / cycle	0.16	0.20	0.24
Average time / gap / cycle			
Captured gap (hrs)	1.20	1.00	0.80
Uncaptured gap (hrs)	1.80	1.50	1.05

In both types of gaps, advanced finishers are assumed to 1) have a higher chance of closing a gap in each work cycle, and 2) require less time per gap per cycle. Moreover, because uncaptured gaps generally require more experience, advanced analysts are assumed to have a relative advantage in uncaptured gaps over captured gaps. Thus, while advanced analysts are assumed to require 67% (0.8/1.2) of the time that beginner analysts require on captured gaps, they require only 58% (1.05/1.8) on a relative basis for uncaptured gaps. The impact of this relative advantage is explored in more detail in Chapter 5.

The modeling parameters of Table 2 correlate with qualitative observations of finisher performance at Whitehead, but are nonetheless artificially precise. Direct use of empirical performance data is difficult because of the collection errors and biases mentioned in previous sections. Instead, we begin our analyses with the assumptions of Table 2, and then explore the sensitivity of our analyses to those assumptions.

3.5 Deriving Simple Results from the Modeling Framework

Using the assumptions of the preceding sections, some simple analytical derivationscan be derived about the probability of single- and multi-gap projects.

3.5.1 Task Completion Probabilities

Let p_N denote the conditional probability that a task is completed during its N^{th} cycle of work, given that it has not been closed during its last N-1 cycles. The probability that the task is completed *on exactly* its N^{th} cycle of work is then:

$$P_N = (1-p_1)(1-p_2)\cdots(1-p_{N-1})p_N$$

If, as this thesis assumes, the conditional probability of closing a gap is the same in each cycle, then $p_1 = p_2 = p_N$ and the above formula simplifies to:

$$P_N = (1-p)^{N-1}p$$

Similarly, the cumulative probability that a task is completed *on or before* its N[th] lab cycle can be derived as the complement of the probability that the task is completed after its N[th] cycle:

$$P_N^* = 1 - \left[(1-p_1)(1-p_2)\cdots(1-p_N)\right]$$

Here again, if we assume that the probability of closing a gap is equal on every cycle, then the above formula reduces to:

$$P_N^* = 1 - (1-p)^N$$

Finally, assuming task independence, the cumulative probability, $Z_N^*(X)$, that a project with X constituent tasks is completed *on or before* its N[th] cycle of work can be derived as the product of the cumulative probabilities of each independent gap.

$$Z_N^*(X) = (P_N^*)^X$$

3.5.2 Expected Task Completion Time

Utilizing the above derivations, we can calculate the expected cycles and time needed to complete a task. The expected number of cycles is just the probability that the task requires N cycles, multiplied by N, and summed over all N. Mathematically, this can be rearranged as follows:

$$ExpectedCycles = \sum_{N=1..\infty} P_N \cdot N = \sum_{N=1..\infty} (1-p)^{N-1} \cdot p \cdot N = \frac{p}{(1-p)} \sum_{N=1..\infty} (1-p)^N \cdot N$$

The last representation is simply p/(1-p) times a power series of the following form:

$$\sum_{n=1..\infty} nx^n = \frac{x}{(1-x)^2}$$

Substituting (1-p) for x in the power series, we arrive at the greatly simplified:

$$ExpectedCycles = \frac{p}{(1-p)} \cdot \frac{(1-p)}{p^2} = \frac{1}{p}$$

Using the above formula with the gap completion probabilities of Table 2, we can estimate the number of cycles required by each finisher skill level for captured and uncaptured gaps. These are indicated in the first two rows of Table 3. Combining these values with our assumptions about the time spent per gap per cycle (also in Table 2), we can further calculate the total expected finisher time required per gap. These are indicated in the second two rows of Table 3.

Table 3. Expected Cycles and Time Per Task.

	Beginner	*Intermediate*	*Advanced*
Average cycles / gap			
Captured gap (cycles)	3.1	2.5	2.1
Uncaptured gap (cycles)	7.1	5.0	3.9
Average total time / gap			
Captured gap (hours)	3.8	2.5	1.7
Uncaptured gap (hours)	13.9	7.5	4.0

3.6 Summary

This chapter has presented data showing the historic variability of Whitehead's finishing process. It has explained why that variability is problematic. Finally, it has provided a modeling framework with which to evaluate the proposals that follow. Though the models reflect actual performance characteristics observed at Whitehead, they are nonetheless simplifications. As such, results based on models should be interpreted in a qualitative light.

4 Task Bundling

Genomes are divided into BACs by means of an enzymatic process, as described in Chapter 2. For a variety of process and technology reasons, Whitehead has historically converted each BAC assembly into a finishing project and then assigned it to a finisher. Because BACs contain a variable number of gaps, however, the volume of finishing work associated with each project assignment may vary significantly. The practice of assigning multiple gaps at a time is referred to as *task bundling* in this chapter.

Task bundling has long been suspected of leading to high output variability, long project cycle times, and a host of other process control problems. Using a model of Whitehead's finishing process, this chapter derives both analytical and simulation-based estimates for the degree to which task bundling aggravates efficient process control. It makes a case for why multi-gap projects should be broken down into single-gap projects before being assigned.

The analysis of this chapter proceeds as follows. First, we analyze the effect that task bundling has on project processing time, showing that when projects contain multiple gaps, their processing time is longer relative to single-gap projects. Next, we use information about how gaps are distributed across BACs to estimate project completion time over a large population of projects. Finally, we use simulation to understand the effect that queuing and lab processing time have on overall project cycle time and process variability. The chapter concludes by presenting a variety of managerial reasons to favor single-gap projects.

4.1 Processing Time for Multi-gap Projects

In this section, we estimate the *processing time* required by multi-gap projects. We define processing time as the active time a finisher spends on a project, as distinguished from *cycle time*, which also includes finisher time, lab time, and queuing delays. Cycle time is considered in more detail in Section 4.3.

To calculate the expected processing time for a multi-gap project, we must calculate the probability that a project takes time T for all $T = 0..\infty$. Consider, for example, the probability that a project containing two captured gaps is completed after four hours of finisher processing (i.e. $T = 4$). We assume, as described in Chapter 3, that each captured gap requires one hour of processing per cycle, undergoes at least one initial cycle of processing, and closes with probability $p = 0.4$ in each cycle.

The probability that our two-gap project requires four hours of work is the cumulative probability of all possible allocations of four hours across those two gaps. The first gap may require one cycle (one hour) of work, while the second gap requires three hours. Conversely, the first gap may require three hours of work while the second gap requires one. A final alternative is that both gaps require two hours of work each. Table 4 shows these allocations.

We can calculate the probability of each of these scenarios by recalling that a captured gap requires N cycles of work with probability $P_N = (1-p)^{N-1} p$. The probability that the first captured gap requires 3 hours is therefore 0.144, while the probability that the other gap requires 1 hour is 0.4. The joint probability of this occurring is 0.144 * 0.4 = 0.058. Completing this analysis for all possible scenarios, we determine that our two-gap project requires four hours of processing with probability P(T=4) of 0.173.

Table 4. Example scenario: probability that a 2-captured-gap project requires 4 hours of processing.

First gap		*Second gap*		*Entire project*	
Work cycles	Probability	Work cycles	Probability	Total time (Hrs)	Joint probability
1	0.400	3	0.144	4	0.058
2	0.240	2	0.240	4	0.058
3	0.144	1	0.400	4	0.058
				Cum. Prob. P(4) =	**0.173**

By performing this same analysis over all time T, we can calculate expected processing time for the project using the following calculation:

$$\text{Expected Processing Time} = \sum_T P(T) * T$$

Extending this analysis to all projects with all possible numbers of captured and uncaptured gaps, we arrive at the following graph of expected project processing times:

Figure 19. Expected processing time for multi-gap projects.

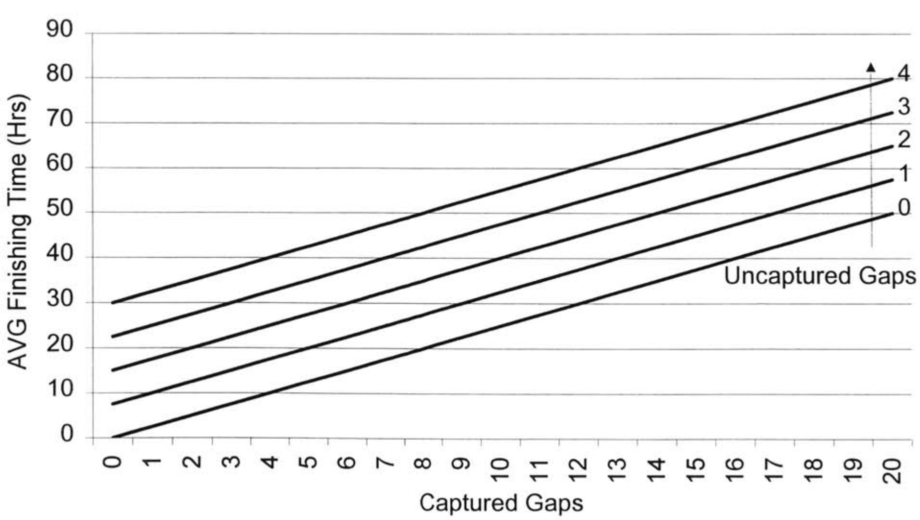


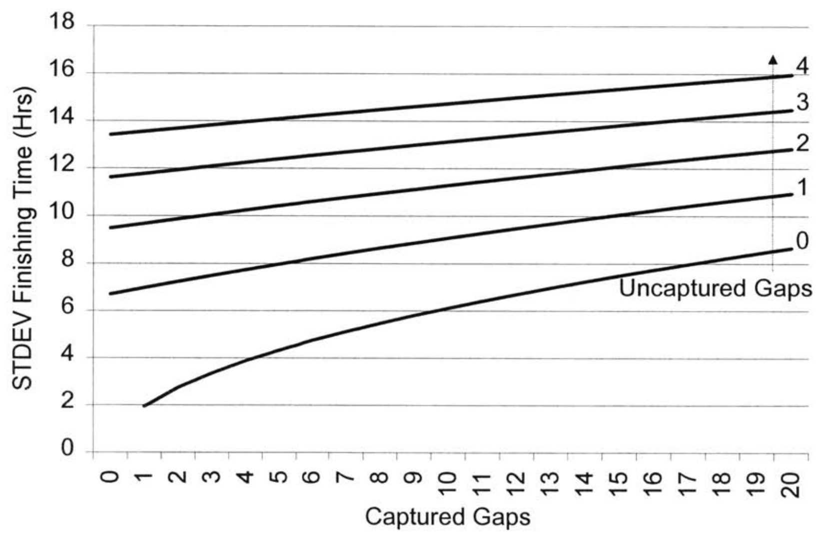
Figure 19 shows that project processing time increases linearly with the number of captured and uncaptured gaps. Each additional captured gap increases expected processing time by 2.5 hours, while each uncaptured gap increases processing time by 7.5 hours. These values correspond exactly to those derived in Chapter 3.

We can extend this analysis to understand the impact of gap bundling not just on expected project processing time, but also on processing time variability. Using the same probabilities calculated above, we can compute the standard deviation of a project's processing time as follows:

$$\text{Standard Deviation of Processing Time} = \sqrt{\sum_T P(T)\cdot T^2 - \left(\sum_T P(T)\cdot T\right)^2}$$

Figure 20 shows standard deviation as a function of a project's captured and uncaptured gap count:

Figure 20. Standard deviation of processing time for multi-gap projects.



While the relationship is not linear, the graph clearly shows that variability increases with gap count. As Figure 21 shows, however, the standard deviation of processing time *as a percentage of the mean* decreases with additional gaps:

Figure 21. Standard deviation of processing time as percentage of mean for multi-gap projects.

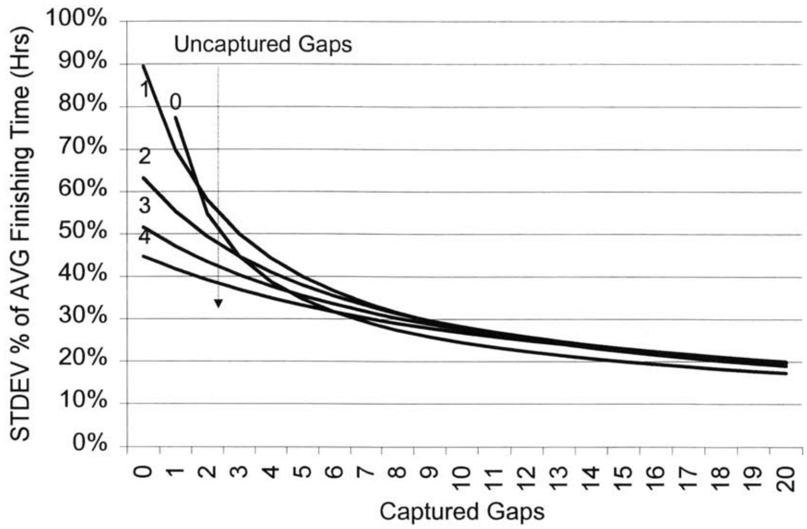


Figure 21 shows that projects with higher gap counts demonstrate less variability relative to their mean than lower gap count projects. This implies an averaging effect. Because gaps are independent, the variability of multiple gaps in a single project tends to cancel out. Exactly how this impacts a finisher's overall productivity is difficult to determine without two additional analyses. In the next section, we synthesize the above analyses with our Poisson model of gap distribution in order to determine the expected processing time of a project selected at random from the larger population. In the section after that, we simulate the finishing workflow in order to account for queuing and lab processing effects.

4.2 Predicting aggregate project lifetimes

In Chapter 3, we made the assumption that captured and uncaptured gaps occur according to Poisson processes with means of 2.1 and 0.5 gaps per project, respectively. By convolving this gap occurrence model with the processing time estimates from the previous section, we can calculate the distribution and mean processing time across all projects in the population.

To calculate the probability that a randomly selected project requires time T to process, we multiply the probability that a project with C captured and U uncaptured gaps occurs by the probability that that the project requires time T, over all C and U. Using $A(X,\lambda)$ to represent the probability of X events in a Poisson process with mean λ, and $P_{C,U}(T)$ to represent the probability that a project with [C,U] gaps requires time T, this is:

$$\text{Prob. that random project takes time T} = \sum_C \sum_U A(C,2.1) * A(U,0.4) * P_{C,U}(T)$$

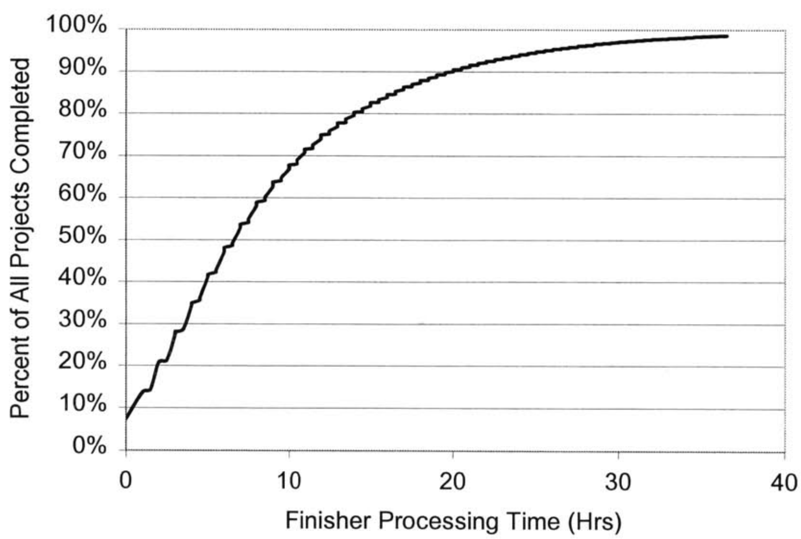
By computing this probability over all time T, we can calculate a distribution of the processing time for all projects in the population. Using this distribution, we can then calculate the mean and standard deviation of the processing time for a randomly selected project. Table 5 compares these values against the mean and standard deviation of several other projects.

Table 5. Mean and standard deviation of processing time for a randomly selected project.

Project	*Mean (Hrs)*	*Stdev (Hrs)*	*Stdev (% of Mean)*
Random project	9.0	8.5	94%
Project with 1 captured gap	2.5	1.9	77%
Project with 1 uncaptured gap	7.5	6.7	89%
Project with 1 cap + 1 uncap gap	10.0	7.0	70%

The analyses indicate that a randomly selected project has a low mean processing time (9.0 hours), but high variability (94% of mean.) If the distribution of project processing time is not symmetric around the mean, we can interpret this high variability to mean that the distribution has a "long tail." In fact, this is confirmed by examining the processing time distribution for a randomly selected project, as shown in Figure 22. Over 80% of projects are completed within 14 hours of processing; however, the last 3% require over 30 hours.

Figure 22. Processing time distribution for a randomly selected project.



Because the finishing workflow is iterative, we expect that the long tail on this distribution would translate into a small population of projects getting "stuck" in a finisher's workflow for many work cycles. We might further suspect that this stickiness would likely heighten a finisher's output variability. To confirm this intuition, we must simulate the effects of queuing and cyclical processing in the finishing workflow. This is the subject of the next section.

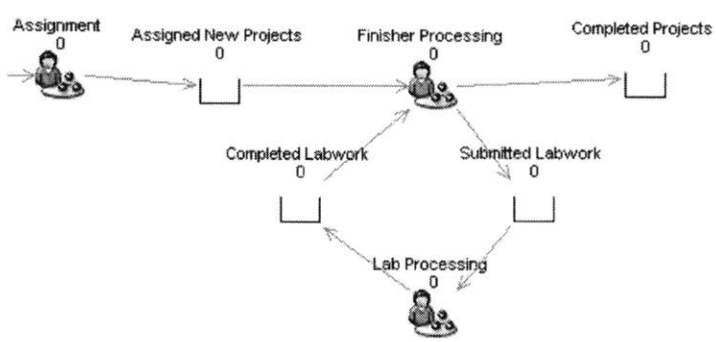
4.3 Impact of Queuing and Cyclical Workflow

In an iterative workflow, analysts must eventually revisit any tasks that they are unable to complete in a given work cycle. As consumers of their own work, any variability present in their output will ultimately impact the rate at which they are able to take on and complete new work. The system is, in effect, a giant feedback loop. In this type of a system, we might expect work to "bunch up" as analysts work their way through random patches of easy and hard projects.

Using Simul8, we model two finishing scenarios in order to understand the effects of queuing and lab processing on finishing process variability. In the first, *Bundled* scenario, captured and uncaptured gaps are inserted into projects according to Poisson processes with means of 2.1 and 0.5 gaps per project, respectively. In the second, *Unbundled* scenario, single-gap projects are inserted into the workflow with captured gap projects occurring 81% of the time and uncaptured gap projects occurring 19% of the time. The relative frequency of the two gap types is therefore identical in both scenarios.[11]

Figure 23 shows our simulated finishing workflow. In both *Bundled* and *Unbundled* scenarios, captured gaps are assumed to require one hour of work per cycle and complete with probability 0.4 at each cycle, while uncaptured gaps were assumed to require 1.5 hours of work per cycle and complete with probability 0.2 during each cycle. Only when all gaps in a project are closed does the finisher deliver the project to his *Completed Projects* queue. Otherwise, the project is submitted for another cycle of lab work. After completing work on a project, the finisher selects his next project from his *Completed Labwork* queue. If no projects are available in that queue, the finisher selected a project from his *Assigned New Projects* queue.[12] Projects are selected out of both queues in strict FIFO order. Lastly, for the purposes of these simulations, an infinite amount of work is assumed to be available for assignment.

Figure 23. Model finisher workflow as represented in Simul8.



[11] Captured gaps occur w. p. 2.1/(0.5+2.1) = 81%; uncaptured gaps occur w. p. 0.5/(0.5+2.1) = 19%.
[12] In Chapter 5, we show that this prioritization strategy ensures that the finisher is fully utilized.

Because of the sensitivity of the system to random number seeds, ten trials are conducted for each of the two scenarios. The simulations are run for 2200 hours (approximately 1¼ years). The first 200 hours of each simulation are considered warm-up and excluded from the results. For each trial, the number of completed projects, captured gaps, and uncaptured gaps are noted at five-hour intervals. These data are then analyzed on a weekly and monthly basis to determine the variability of output.

Table 6 shows the results of these test trials. For each time base, the mean and standard deviation of output is measured for projects, captured gaps, and uncaptured gaps.

Table 6. Simulated results comparing bundled and unbundled task assignment.

	Bundled	*Unbundled*
Weekly output		
Project completions		
Mean rate (projects/week)	3.6	10.1
Standard deviation of rate (% mean)	53%	26%
Captured gap completions		
Mean rate (gaps/week)	8.3	8.3
Standard deviation of rate (% mean)	61%	31%
Uncaptured gap completions		
Mean rate (gaps/week)	1.7	1.9
Standard deviation of rate (% mean)	70%	58%
Monthly output		
Project completions		
Mean rate (projects/week)	14.4	40.5
Standard deviation of rate (% mean)	28%	14%
Captured gap completions		
Mean rate (gaps/week)	33.3	33.0
Standard deviation of rate (% mean)	30%	16%
Uncaptured gap completions		
Mean rate (gaps/week)	7.0	7.4
Standard deviation of rate (% mean)	45%	34%

Note that by all measures, on either a weekly or monthly basis, finisher throughput is nearly identical under both scenarios. (The reader should compare captured and uncaptured gap completion rates; project completion rates are not directly comparable because projects in two scenarios have different numbers of gaps.) Output variability, however, is substantially reduced in the Unbundled scenario. On weekly and monthly bases, captured gap output variability is nearly halved. Uncaptured gap output variability is also reduced.

With these simulated results, we confirm what preceding analyses suggested – that breaking multi-gap projects down into single-gap projects can substantially reduce the variability of a finisher's output. Together with shorter project processing times, as demonstrated earlier in the chapter, *task unbundling* promises more predictable, manageable workflows.

4.4 Managerial Reasons to Favor Unbundling

Even without these analytical arguments, there are a host of other managerial reasons to favor task unbundling. One natural consequence of projects that require longer, more variable processing time is that analysts must maintain higher WIP levels. High WIP levels, however, frustrate efficient process monitoring and control. Analysts with large queues are often tempted to process work selectively according to short-term objectives. In Chapter 5, this *WIP gaming* phenomenon is proven detrimental not only to analysts' individual performance, but also to the stability of the group's output.

Perhaps more importantly, bundling multiple tasks into a project obfuscates the state of individual tasks, leaving management powerless to identify problematic tasks. At Whitehead, for example, there are formal mechanisms for tracking projects, but individual gaps are not tracked in a standard, disciplined manner. As a result, management's ability to identify and triage problematic tasks is hampered. Similarly, because management has limited insight into the status of individual tasks, it cannot measure analysts' aptitude with individual tasks. It is also difficult for management to identify best practices from among the many methods employed by finishers. In short, task bundling obscures management's insight into the real unit of work – the task.

Thus, there remain significant process control reasons – beyond the analytical arguments presented earlier in this chapter – to warrant task unbundling. Analysts managing discrete, single-task projects are less likely to game their workload. Moreover, unbundled tasks are easier to monitor and, as such, pave the road towards the identification and development of best practices.

4.5 Summary

This chapter has shown that task bundling, while a natural artifact in some data analysis workflows like finishing, unnecessarily inflates project-processing times, boosts WIP levels, and increases process variability. From a management perspective, bundling also hides important process details and impedes organizational learning. To the extent possible, then, managers should consider policy changes that enable the delivery of single-task projects to analysts.

The chapters that follow start from this assumption and probe more deeply into other important workflow policies.

5 Task Assignment

This chapter examines how to optimally assign tasks to a group of analysts. Task assignment can be broken down into two basic policy questions: *timing policy* defines when new tasks should be assigned to analysts; and *routing policy* defines which tasks should be assigned to which analysts. The first half the chapter deals with timing, demonstrating that a just-in-time assignment policy ensures 100% analyst utilization, low variability, and reduced work-in-process. The second half of the chapter deals with routing, showing that, under certain circumstances, assigning complex tasks to senior analysts yields higher group productivity. Implementation of both policies can ensure high utilization and optimal productivity in analysis workflows like finishing.

5.1 Timing of New Task Assignments

Analysts are a limited resource, requiring significant time to hire and train. As such, analysis groups like Whitehead's Finishing Group often prove to be a major bottleneck in the production process. Prompt assignment of new tasks is critical to ensuring that analysts stay fully utilized. At the same time, overly aggressive task assignment can drive up WIP levels, lengthen project cycle times, and increase process variability. Proper timing of new task assignments can ensure high analyst utilization without these negative side effects.

Whitehead has historically utilized an *on-demand* policy, in which analysts solicit new tasks from their managers at will. The productivity implications of this policy are comparedto two potential alternatives: *kanban* assignment, in which analysts are fed a new task whenever they complete a previously assigned task; and, *just-in-time* assignment, in which analysts are assigned a new task whenever they have no other tasks to process. Just-in-time assignment is shown to ensure 100% utilization and low WIP levels. Moreover, it is simple to implement and eliminates analysts' tendency to selectively prioritize their workload – a behavior that increases process variability.

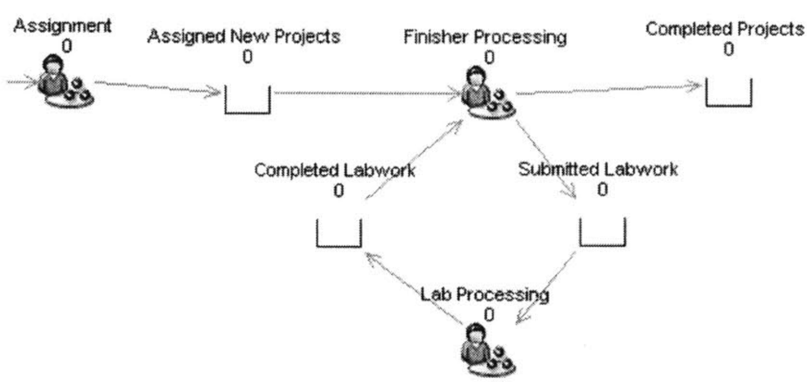
In the following analyses, a simplified model of the finishing workflow is assumed as described in Chapter 3. Building on the analyses of Chapter 4, projects are assumed to consist of one gap each, with captured gap projects and uncaptured gap projects occurring 81% and 19% of the time, respectively. Simulations were conducted in Simul8. The chief metrics for comparison between the policies are finisher utilization (working time divided by elapsed time), output variability (standard deviation of output on a weekly basis), and project cycle time (time between assignment and completion). For each policy, ten trials of 2200 hours each were conducted. The data presented represents results averaged across all trials. Finally, the model assumes that there is no shortage of projects available for assignment.

The following table summarizes finisher performance assumptions used in the simulations of this chapter (these assumptions are copied from Chapter 3):

Table 7. Finisher performance characteristics used in simulations.

Gap closing probabilities	
Captured gap / cycle	0.4
Uncaptured gap / cycle	0.2
Average time / gap / cycle	
Captured gap (hours)	1.0
Uncaptured gap (hours)	1.5
Average cycles / gap	
Captured gap (cycles)	2.5
Uncaptured gap (cycles)	5.0
Average total time / gap	
Captured gap (hours)	2.5
Uncaptured gap (hours)	7.5

The diagram below shows the basic workflow for an individual finisher. New projects are placed into the Assigned New Projects queue according to the given task assignment timing policy. Projects that have been processed by the lab are returned to the finisher's Completed Labwork queue. The finisher can select his next task from either of these two queues, as determined by the task assignment timing policy.



5.1.1 Status Quo: The On-Demand Model

Historically, Whitehead has followed an *on-demand* task assignment policy in which finishers are allowed to request new projects at will. Modeling this policy is difficult because, in practice, each finisher follows different criteria for requesting new work. In the simulations that follow, we assume that our model finisher follows a simple rule-of-thumb behavior: whenever the number of projects in his Completed Labwork queue falls below some watermark, he requests a new project and begins work on it immediately.[13]

The table below reflects the utilization, variability, and project cycle times across a variety of these watermark levels. (Note that a watermark of zero is not shown, as it is essentially the *just-in-time* assignment model analyzed in a subsequent section.)

Table 8. Simulation results for on-demand task assignment.

	Watermark Level		
	3	**6**	**9**
Finisher utilization (%)	100%	100%	100%
Average weekly output (projects/week)	10.4	10.5	10.6
Average weekly output variability (stdev/avg)	26%	26%	26%
Active finisher time per project (hours)	3.45	3.45	3.45
Lab Submissions queue			
Queue delay on each cycle (hours)	70.2	70.2	70.2
Total delay over all cycles (hours)	138.6	138.6	138.6
Completed Labwork queue			
Queue size (projects)	2.7	5.6	8.5
Queue delay on each cycle (hours)	4.8	9.8	15.0
Total delay over all cycles (hours)	9.5	19.4	29.6
Total project cycle time (hours)	151.5	161.5	171.6

As the table illustrates, the performance of each finisher – in terms of output, variability, and utilization – is nearly identical for all watermark levels. The critical difference between each lies in queue levels and project cycle time. Finishers using a watermark of 9 projects have an average of 8.5 projects in their Completed Labwork queue and an associated queuing delay of 15 hours.

To see how this added queuing delay translates into higher total project cycle time, note that cycle time is the sum of finisher processing time, lab processing time, and queuing delays between the finisher and the lab. Finisher processing time can be calculated as the expected time the analyst spends on either of the two gap types:

Capgaps/Project * Time/Capgap + Uncapgaps/Project * Time/Uncapgaps = Time/Project

81% * 2.5 + 19% * 7.5 = 3.45 hours/project

[13] This model overlooks the fact that finishers' behaviors tend to change over time. Also, coordinators can decline a finisher's request if they feel the finisher already has sufficient work. Nonetheless, the model does reflect finishers' oft-stated desire to not wait until they have nothing to do before requesting new work. Finishers also tend to respond to their Completed Labwork queue, since it is the most visible indicator of how much work they have to do in the near future.

To determine lab processing time and queuing delays, we must first know how many times a project is cycled between the finisher and the lab. In the finisher workflow, projects are processed once initially without labwork. Captured gap projects receive an average of 1.5 cycles of lab work, while uncaptured gap projects receive 4.0 cycles of lab work. With a ratio of 81:19 between captured and uncaptured gap projects, the average project is sent to the lab 81%*1.5 + 19%*4 = 1.98 times.

We assume the lab begins processing labwork immediately. Delays due to the Submitted Labwork queue are therefore zero. The lab requires an average of 70 hours to perform its work. Average total delays due to lab processing are therefore 70 * 1.98 = 138.6 hours. Finally, as determined by the simulations above, with a watermark of 9, projects spend an average of 15 hours in the Completed Labwork queue for each lab cycle. Total average delay for a project due to the Completed Labwork queue is therefore 15 * 1.98 = 29.6 hours.

Total project cycle time for a watermark of 9 is then: 3.5 of hours finishing time plus 138.6 hours lab processing time plus 29.6 hours queuing, for a total of 171.6 hours.

Comparing this against watermarks of 3 and 6 projects, it is clear that higher watermarks mean higher queue levels and longer project cycle times. High queue levels have a number of disadvantages, most notably the temptation they create for finishers to selectively prioritize their work. These issues are discussed in more detail in Chapter 6.

Beyond these issues, the on-demand model is problematic mostly because it is so varied and unpredictable. At Whitehead, finishers followed a wide range of personal policies. Some maintained fairly lean work queues (i.e. low watermarks), while others stockpiled large numbers of projects. From a management perspective, this added unnecessary uncertainty to the system. It was difficult to estimate how long projects would take to complete. A project might be delayed because it was particularly difficult; on the other hand, it might be delayed simply because its finisher maintained a large work queue.

5.1.2 Kanban Assignment

In the kanban model, the number of projects an analyst can possess at any point in time is limited. Once an analyst has reached this limit, he or she can only acquire a new project after completing an in-process project. In the model below, whenever our model analyst acquires a new project, he begins work on it immediately before returning to other work.

A theoretical kanban limit enabling 100% finisher utilization can be calculated as the ratio between the average elapsed time and active time required by a project. For example, if it takes an average of 100 hours to move a project through the system, but our model finisher performs only 2.5 hours of work on each project, he would require at least 100/2.5 = 40 tasks to stay fully utilized.

In the finishing example provided, we have already calculated the average finisher time required per project as 3.5 hours. Elapsed time is more difficult to calculate since it depends on queuing. Ignoring queuing delays, the elapsed time is just the sum of finishing time and lab processing time. In the previous section, lab processing time was calculated as 138.6 hours This yields an expected project cycle time, excluding queuing, of 138.6 + 3.5 = 142.1 hours. Ignoring queuing, then, a theoretical lower bound for the number of projects a finisher must have in order stay fully utilized is 142.1 / 3.5 = 41 projects.

Table 9 compares the simulated performance of a kanban level of 41 with a variety of other kanban levels:

Table 9. Simulation results for kanban task assignment.

	Kanban Size				
	30	**35**	**41**	**45**	**50**
Finisher utilization (%)	69%	79%	90%	96%	99%
Average weekly output (projects/week)	7.3	8.4	9.6	10.1	10.5
Average weekly output variability (stdev/avg)	47%	42%	34%	32%	27%
Active finisher time per project (hours)	3.5	3.5	3.5	3.5	3.5
Lab Submissions queue					
Queue delay on each cycle (hours)	70.3	70.2	70.3	70.3	70.3
Total delay over all cycles (hours)	138.8	138.7	138.6	138.6	138.5
Completed Labwork queue					
Queue size (projects)	0.6	1.2	2.3	3.9	7.3
Queue delay on each cycle (hours)	1.6	2.6	4.6	7.1	12.9
Total delay over all cycles (hours)	3.1	5.0	9.0	14.0	25.4
Total project cycle time (hours)	145.3	147.2	151.0	156.0	167.4

The results indicate the system is extremely sensitive to proper setting of the kanban level. The theoretical lower bound of 41 projects proves insufficient to maintain high finisher utilization (90%.) As a result, output suffers both in terms of total output (9.6 projects/week) and variability (34%) relative to higher kanban levels. Clearly, the kanban limit must be set large enough to account for queuing delays. Increasing the limit to 50 projects proves sufficient to achieve nearly 100% utilization, higher output (10.5 projects/week), and low variability (27%).

However, this improved utilization comes at the cost of increased queue size. The finisher has an average of 7.3 projects in his Completed Labwork queue. The delay each project experiences due to this queue is 25.4 hours, yielding a project cycle time of 167.4 hours. This represents an 18% increase over the 142 hours we estimated for a system with no queuing delays.

Next, we consider the other extreme. Reducing the kanban limit too much can have deleterious effects on the finisher's productivity. Had the kanban limit been set to 30 projects, for example, output would have fallen to 7.3 projects/week with a variability of 47%.

In short, the efficiency of a kanban-based assignment policy depends critically on setting the kanban limit correctly. Since precise determination of an analyst's capacity is difficult when task mix and skill level vary so significantly, this limit is nearly impossible to set incorrectly. The negative effects of setting the kanban limit incorrectly are significant. If it is set too high, large WIP levels result. If it is set too low, finisher efficiency falls precipitously.

5.1.3 Just-in-time Assignment Policy

In a just-in-time policy regime, an analyst is assigned a new project only when he or she has no projects to be processed in his or her Completed Labwork queue. By definition, then, a just-in-time policy ensures 100% utilization since a finisher always has a project to process.[14] The performance of this policy is compared against the top-performing kanban scenario in the table below:

Table 10. Simulation results for just-in-time task assignment.

	JIT	*Kanban-50*
Finisher utilization (%)	100%	99%
Average weekly output (projects/week)	10.5	10.5
Average weekly output variability (stdev/avg)	27%	27%
Active finisher time per project (hours)	3.45	3.45
Lab submissions queue		
Queue delay on each cycle (hours)	70.2	70.2
Total delay over all cycles (hours)	138.6	138.5
Lab returns queue		
Queue size (projects)	0.9	7.3
Queue delay on each cycle (hours)	1.6	12.9
Total delay over all cycles (hours)	3.1	25.4
Total project cycle time (hours)	145.1	167.4

In all productivity-related factors, a just-in-time policy is comparable to the highest performing kanban (50 project limit) model: the finisher maintains 100% utilization, has an output of 10.5 projects/week, and an output variability of 27%. However, the just-in-time policy accomplishes this without the increased WIP, queuing delays, and longer project cycle time associated with the kanban model.

[14] This is true as long as new projects are available for assignment, as assumed in these models.

Moreover, from a process management perspective, a just-in-time policy is far easier to implement than a kanban model. It requires no a priori knowledge about a finisher's capacity, nor does it create the risks of underutilization or high WIP levels that are associated with incorrectly estimating a finisher's capacity, as in the kanban model.

5.1.4 Assignment Timing Policies: Conclusions

Just-in-time task assignment enables 100% finisher utilization, while ensuring low WIP levels and minimal project cycle times. Allowing analysts to request new tasks on their own can lead to increased WIP levels and process variability. Likewise, the kanban model, while attractive for its apparent discipline, is difficult to implement correctly, and costly when it is done wrong. A just-in-time assignment policy enables high analyst utilization, is trivial to implement, and ensures low WIP levels.[15]

5.2 Routing of New Assignments

Having identified a timing policy that ensures 100% analyst utilization, the question now becomes one of how to optimally route tasks to a group of analysts. In the model below, there are two types of tasks and three analyst skill levels as described in Chapter 3. The analysis shows that it is only optimal to route difficult tasks to senior analysts when they possess a relative performance advantage in those tasks over junior analysts. Thus, if senior analysts are three times as fast as junior analysts in processing simple tasks, they must be *at least* three times as fast in the difficult task to warrant being assigned difficult tasks.

5.2.1 Linear Optimization Setup

Using the linear optimization routine in Microsoft Excel's Solver, optimal task allocation was explored for a finishing group consisting of two analysts at each of three skill levels: beginner, intermediate, and advanced. The planning horizon for task allocation was 3600 hours, or approximately 2 years. An infinite supply of the two basic task types (captured and uncaptured gaps) was assumed to exist. Tasks can be allocated in any way to the analysts. However, the group must collectively complete captured and uncaptured gaps in the same ratio that they naturally occur (i.e. 81% captured and 19% uncaptured.)

The average gap completion rate for each analyst skill level is shown in Table 11. The relative performance of each skill level, normalized to the intermediate skill level, is indicated in

[15] In the analyses of this thesis, we assume that new work is always available for assignment. In the genomics context, this is a reasonable assumption because BACs are produced, sequenced, and assembled within a short timeframe at the outset of a genome. Additional work is needed to qualify the results of this thesis in contexts where new work is not consistently available for assignment.

the last two rows. (Note that these numbers were first derived in Chapter 3. Gap completion rates are calculated as the reciprocal of the average time per gap, as calculated in Table3.)

Table 11. Gap completion rates and relative performance by finisher skill level.

	Beginner	***Intermediate***	***Advanced***
Average gap completion rate			
Captured gap (gaps/hour)	0.27	0.40	0.60
Uncaptured gap (gaps/hour)	0.09	0.13	0.23
Relative performance v. intermediate			
Captured gap	67%	100%	150%
Uncaptured gap	67%	100%	171%

The two figures that are of particular importance in this analysis are highlighted in gray. Advanced analysts are assumed to have a slight *relative* advantage in uncaptured gaps over beginner and intermediate analysts. The advanced analysts are faster than their peers in either task, but they are proportionately faster when processing uncaptured gaps. They process captured gaps 50% faster than intermediates, but can process uncaptured gaps 71% faster.

5.2.2 Optimal Task Allocation

To determine the benefits of optimal task allocation in this (or any other) system configuration, we must first establish a baseline performance. If we assume that there were no advantages to preferential task allocation, then by default we would distribute tasks randomly to analysts regardless of skill. The output for the group over a period of two years is shown in Table 12. Because task distribution is random, 81% of each skill groups' output is captured gaps, while the remaining 19% is uncaptured gaps.

Table 12. Productivity of random task distribution.

	Beginner	*Intermediate*	*Advanced*	**Total Gaps**
Captured gaps completed	1132	1699	2688	5519
Uncaptured gaps completed	270	404	640	1314
Total gaps completed	1402	2103	3328	6833

To understand how this system might be optimized, we formulate a linear program (LP). Six finishers (two from each skill level) are assumed to work for a period of two years. They complete captured and uncaptured gaps according to the skill-based rates indicated in Table 11. Gaps may be allocated in any manner to the six finishers. However, the group's total output over the two years must consist of 81% captured gaps and 19% uncaptured gaps. The goal of our LP is to maximize the total number of gaps completed, subject to this gap-mix constraint.

Table 13 shows the group's optimal output as determined by this LP formulation. An output of 7026 projects shows that there is, in fact, a more optimal allocation of tasks relative to the random distribution model, which completed only 6833 projects. Studying the allocation of gaps in Table 13, we see that it is optimal to allocate uncaptured gaps exclusively to advanced analysts, while all skill levels contribute to the completion of captured gaps.

Table 13. Productivity under optimal task distribution.

	Beginner	*Intermediate*	*Advanced*	**Total Gaps**
Captured gaps completed	1941	2912	821	5675
Uncaptured gaps completed	0	0	1351	1351
Total gaps completed	1941	2912	2172	7026

5.2.3 Sensitivity Analysis

A closer analysis shows that the optimal task allocation derived above is sensitive to the advanced analysts' relative performance advantage in uncaptured gaps over captured gaps. Mathematically, this relative advantage can be expressed as the ratio of the advanced analysts' performance advantage in uncaptured gaps to their performance advantage in captured gaps, or:

$$\frac{UncapturedRate_{Advanced} / UncapturedRate_{Beginner}}{CapturedRate_{Advanced} / CapturedRate_{Beginner}}$$

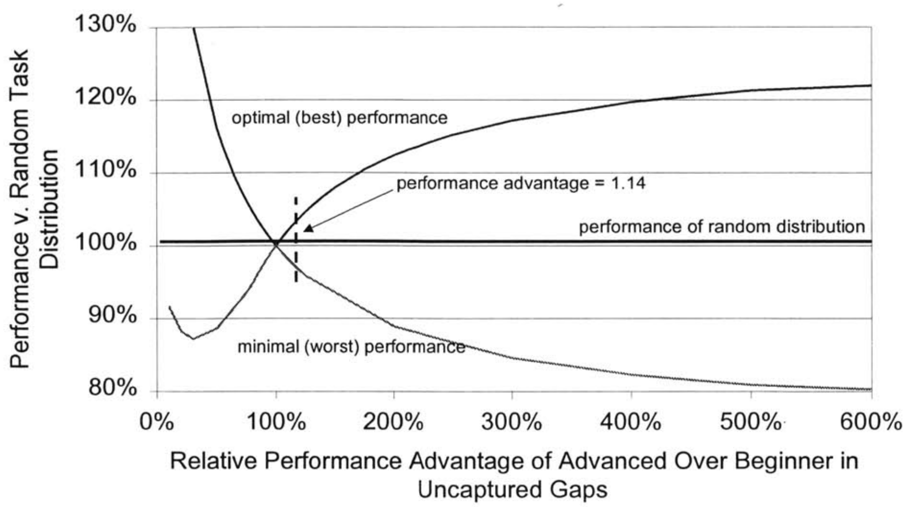
In the system configuration being considered, the advanced analysts' relative performance advantage in uncaptured gaps over beginner analysts is therefore:

$$\frac{0.23/0.09}{0.60/0.27} = 1.14$$

By varying our estimate of the advanced analysts' rate of uncaptured gap completion (initially, 0.23 gaps/hour), we can explore the sensitivity of the optimal solution just derived.

The graph below shows how the group's performance with optimal task allocation varies according to advanced analysts' relative performance advantage. The top curve represents the performance of optimal allocation relative to random distribution; the bottom curve represents the minimum (i.e. worst possible) task allocation. The dotted line represents the case just discussed (a performance advantage of 1.14.)

Figure 24. Performance gains from optimal task distribution.



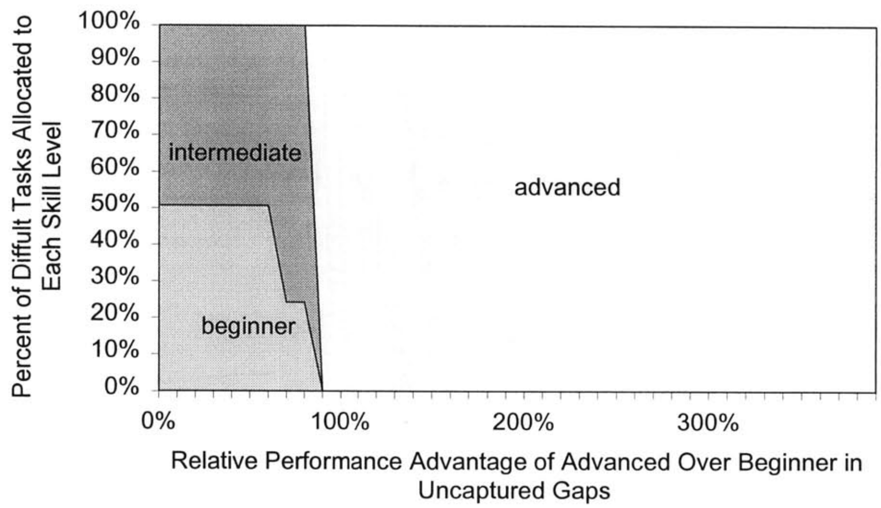
Note that the gains from optimal task allocation versus a random approach increase as the advanced analysts' performance advantage increases. Thus, a relative performance advantage of 300% means that the group's output could be improved by over 15% through optimal task allocation. Conversely, the group's output could be reduced by over 15% if tasks were allocated incorrectly.

Note, too, that when advanced analysts possess no relative advantage in uncaptured gaps over beginner analysts (i.e. X=100% in the above graph), there are no gains from optimal task allocation. In this case, any task distribution yields optimal performance. Random distribution would likely be preferred since it is easiest to implement.

Finally, note that when advanced analysts are at a performance *disadvantage* on uncaptured gaps (to the left of X=100% in the graph above), it is still possible to achieve performance in excess of random task distribution. These gains are achieved, however, by allocating uncaptured gaps to beginner and intermediate analysts.

The next graph shows the relative portion of uncaptured gaps allocated to each skill group as it varies according to the advanced analysts' relative performance advantage in uncaptured gaps. When advanced analysts possess a relative advantage, they are assigned all of the uncaptured gaps. When they are at a disadvantage, however, uncaptured gaps are allocated exclusively to intermediate and beginner analysts.

Figure 25. Optimal allocation of uncaptured gaps between skill levels.



5.2.4 Task Routing Policies: Conclusions

It is tempting to assume that difficult tasks should preferentially be assigned to more highly skilled analysts. In fact, this is true only when advanced analysts are faster in those difficult tasks relative to other tasks. If the advanced analysts possess no relative advantage, there are no benefits to preferential task allocation and random distribution will suffice. If, however, advanced analysts perform at a relative disadvantage in difficult tasks, then preferentially assigning them can actually *reduce* group productivity versus a random distribution approach.

In Whitehead's Finishing Group, experienced finishers almost certainly possess relative performance advantages in some of the more difficult tasks. Beginner finishers are often unable to complete more complex tasks without assistance. Thus, in these circumstances, preferential assignment of difficult tasks to experienced finishers can result in higher group productivity. Nonetheless, management must scrutinize finishers' relative performance across the spectrum of tasks to determine if and when these relative advantages disappear. Continuing the preferential assignment of difficult tasks to experienced finishers may eventually harm group productivity.

Finally, note that we have ignored the training value that beginner and intermediate finishers derive from working on difficult tasks. Doing so may reduce short-term productivity but boost long-term productivity by giving finishers important learning opportunities. This issue is studied in more detail in Chapter 8.

5.3 Summary

Proper timing and routing of new tasks is critical to maintaining high analyst utilization, low WIP levels, and optimal group performance. A *just-in-time* timing model ensures high analyst utilization without the high WIP and long cycle times associated with Whitehead's existing policy or a kanban-based approach. Preferential routing of difficult tasks to advanced analysts was shown to be a performance enhancer, but only when advanced analysts possess a relative performance advantage in those tasks.

6 Workload Management

It is tempting for the manager of a complex data analysis process to grant analysts broad discretion in how they manage their workload. Indeed, some would argue that doing so is critical to giving analysts a sense of empowerment and ownership of their work. Through simulation and anecdotal evidence gathered at Whitehead, this chapter shows that the advantages of granting analysts broad discretion must be balanced against the process control problems that it creates. Specifically, analysts that function with few constraints tend to prioritize work according to short-term, sub-optimal objectives. Doing so increases process variability and wreaks havoc with management's ability to manage and predict the group's productivity. This chapter illustrates the importance of enforcing a first-in-first-out (FIFO) workflow order.

6.1 The Temptation to "Harvest"

Large WIP levels and a discretionary workflow create conditions under which both management and analysts will be tempted to reprioritize work according to short-term objectives. Without proper controls, this reprioritization tends to favor simpler tasks because they can be completed more easily and boost short-term productivity. This phenomenon of favoring simpler tasks is referred to as "harvesting" in this chapter. When large work queues build up, analysts *harvest* simpler tasks out of those queues to meet short-term goals.

Harvesting rarely boosts long-term productivity. Rather, it simply moves easy tasks forward, while deferring complex tasks. At best, this reprioritization is a wash in terms of net productivity. More likely, however, reprioritization will have negative long-term effects on the process. It creates a false sense of progress, it makes the process harder to predict, and it results in a buildup of difficult future work. If the harvesting is achieved through short-term initiatives, efficiency may also be lost to the overhead associated with those initiatives. Finally, there are psychological effects: as tasks become increasingly difficult, analysts and management may become increasingly frustrated with their slow progress.

Analysts and management both play important roles in harvesting. For analysts, the temptation to harvest is driven by a variety of short-term, often personal objectives. The analysts may be frustrated by an influx of difficult tasks; thus, they seek temporary respite in easier tasks. Or, their performance reviews may be imminent, giving them an incentive to temporarily boost output by deferring complex tasks. Just as analysts are tempted to harvest to meet short-term objectives, so too is management. When production deadlines loom, management may be tempted to pressure analysts to boost their output. Without specific guidelines about how that boost should be achieved, analysts inevitably turn to harvesting. Thus, while analysts do the actual harvesting, management is often complicit in the behavior.

These phenomena were readily apparent in Whitehead's Finishing Group, particularly near the end of the Human Genome Project. Finishers were allowed to carry workloads many times in excess of their true capacity. In response to looming HGP deadlines, management implemented a variety of short-term initiatives to boost output. Because finishers carried large workloads, they had the flexibility to dip into their queues, find easy tasks, and complete them in order to meet management's goals. Over time, however, their ability to do so diminished. As the HGP neared completion in early 2003, all that remained were difficult tasks. Management and analysts grew increasingly frustrated with their inability to boost or even predict output. The price for harvesting had come due.

6.2 Simulations

The following simulations show that the combination of large work queues and a discretionary workload management policy results in higher output variability and long-term caching of difficult tasks in the workflow. Two contrasting workflow policies are explored: *FIFO* and *Harvest*. In the FIFO policy, finishers process tasks in the order that they are returned from the lab. In the Harvest policy, finishers survey their queue of tasks returned from the lab and process captured gaps first (before uncaptured gaps) because they require less work to complete.

Like previous chapters, we assume that projects contain either a single captured gap with probability 0.81 or a single uncaptured gap with probability 0.19. Captured gaps require one hour of work per cycle and close with probability 0.4 in each work cycle. Uncaptured gaps require one and a half hours of work per cycle and close with probability 0.2.

Both the FIFO and Harvest models are explored using three task assignment configurations. In the first configuration ("just-in-time"), our model finisher is assigned tasks only when he has no tasks in his lab completions queue. In the second configuration ("kanban-50"), the finisher is limited to having 50 projects in circulation at a time. In the third configuration ("kanban-75"), the finisher is limited to having 75 projects in circulation. We examine the three task assignment configurations in order to understand the potential interactions between the FIFO and Harvest workload policies and the level of WIP in the system. From the previous chapter, we know that the just-in-time configuration has the least WIP, while the kanban-75 configuration has most.

The following table shows the results of simulating the FIFO and Harvest policies in each of the three task assignment configurations. Twenty trials were conducted for each configuration, with results averaged across the trials. The simulations were run for 2200 hours (approximately 1¼ work years.) The first 200 hours were considered warm-up and are excluded from the figures below. The first set of rows in the table represents raw finisher output; the second set shows output variability (σ/μ); the third set shows average WIP maintained by the finisher; finally, the last set shows project cycle times.

Table 14. Workload management policy simulations.

	Harvest			***FIFO***		
Task assignment configuration	JIT	kanban-50	kanban-75	JIT	kanban-50	kanban-75
1. Completed Projects	645	655	686	647	649	658
2. Captured Gap Projects (% total)	82%	83%	85%	82%	82%	83%
3. Uncaptured Gap Projects (% total)	18%	17%	15%	18%	18%	17%
4. Project output variability (σ/μ, weekly)	27%	31%	31%	26%	28%	27%
5. Captured Gap output variability	32%	37%	38%	31%	35%	32%
6. Uncaptured Gap output variability	60%	60%	70%	59%	60%	61%
7. Work-in-process (WIP) Projects	44	48	73	44	49	73
8. Captured Gap Projects (% total)	62%	56%	42%	63%	63%	64%
9. Uncaptured Gap Projects (% total)	38%	44%	58%	37%	37%	36%
10. Average Project Cycle Time (hours)	149	164	232	150	165	242
11. Captured Gap Project (% of avg)	77%	70%	52%	78%	78%	80%
12. Uncaptured Gap Project (% of avg)	192%	224%	308%	190%	190%	182%

Looking first at the finisher's output, we see that the finisher is able to complete more projects as his WIP levels are increased. Only 645 projects are completed in the JIT configuration, whereas 655 and 686 projects are completed in the kanban-50 and kanban-75 configurations. Examining the captured/uncaptured gap mix (rows 2-3), however, we see that the finisher did so by increasing his output of captured gaps. Captured gaps are easier; by favoring them, we would expect the finisher's output to increase. The FIFO model, on the other hand, maintains nearly constant project output and captured/uncaptured gap mixes, regardless of the task assignment configuration.

Looking next at output variability (rows 4-6), we see that the finisher's weekly variability increases with his WIP level in the Harvest policy regime. From JIT to the kanban-75 configuration, the variability of captured gap output increases from 32% to 38%, while the variability of uncaptured gap output increases from 60% to 70%. Output variability in the FIFO configurations, however, remains unaffected by the increased WIP levels. From this, we can deduce that it is the combination of *both* increased WIP levels *and* discretionary workflow policies like the Harvest model that lead to increased output variability.

The next set of data (rows 7-9) shows that the Harvest and FIFO policies demonstrate similar WIP level increases as the kanban limit is raised. However, in the case of the Harvest policy, the relative portion of uncaptured gaps in the WIP increases. In the JIT configuration, 38% of WIP projects are uncaptured gap projects. This rises to 62% in the kanban-75 configuration. This confirms what we should already have suspected from the finisher's output. By favoring captured gaps in his workflow, the finisher ends up caching large numbers of uncaptured gaps. In the FIFO model, where this form of discretion is not permitted, the finisher's mix of uncaptured and captured gaps remains constant regardless of the WIP levels.

Examining rows 10-12, we see finally that caching of uncaptured gaps translates into increased cycle times for those projects. In the FIFO policy regime, cycle times for all projects increase with WIP level, but the increase is proportionately the same for captured and uncaptured gap projects. Relative to the average project cycle time, uncaptured gap projects take roughly twice as long (190%) regardless of the kanban limit. For the Harvest policy regime, however, we see that caching uncaptured gaps results in higher and higher queuing delays for those projects. Whereas uncaptured gaps take twice as long as the average project in the JIT configuration (190%), they take three times as long in the kanban-75 configuration (310%). The cycle time for captured gaps, on the other hand, falls relative to the average project cycle time. In summary, the combination of high WIP levels and preferential treatment of captured gaps results in significantly higher cycle times for uncaptured gaps. For the finishing example considered here, this means that a captured gap takes on average 3 weeks to pass through the system, while an uncaptured gap takes almost 20 weeks!

One additional observation worth noting is that both the FIFO and Harvest policies perform similarly when tasks are assigned in a "lean" JIT manner. Productivity is high, output variability is low, and the WIP task mix remains constant. These results show that when analysts are prevented from building up large work queues, the potential negative effects of harvesting are virtually eliminated. For managers considering implementation of the FIFO policy, these results reveal a convenient alternative: rather than force FIFO ordering on analysts, who may view the policy as draconian, management may be able to eliminate harvesting and its effects through JIT task assignment.

6.3 Managing Exceptions to FIFO Workflow

The preceding section showed that a FIFO workflow policy reduces output variability and eliminates the caching of difficult tasks. While attractive for these reasons, the FIFO policy also has some practical limitations. Specifically, analysts may find themselves unable to complete a task if it proves too difficult for their skill set. In a FIFO policy regime, however, analysts would be forced to continue processing these tasks ad infinitum, long after it became clear that they would not be able to complete them. Clearly, for the FIFO policy to work, analysts must have a procedure (an "escape hatch") by which they can pull a stalled task out of their workload.

In the Harvest policy regime that existed at Whitehead, analysts had the discretion to defer stalled tasks and thereby focus on other, more productive tasks. Allowing analysts to defer stalled tasks in a FIFO policy regime, however, would effectively turn it into a discretionary, Harvest-like workflow policy. Difficult tasks would be deferred at will, resulting in larger and larger queues of untreated, problematic tasks. Instead, what we seek is an explicit, institutionalized process by which analysts can escalate stalled tasks and receive assistance from senior personnel. Having such a policy ensures that tasks will not stall the FIFO workflow and will be handled in a timely manner.

One solution we might consider is a policy in which incomplete tasks are reviewed with senior personnel after a certain time period or number of work cycles. Until that time has elapsed, analysts are required to continue processing the task in FIFO order. If the task proves difficult and eventually stalls, then a forced periodic review will give the analyst the opportunity to seek advice on the task. An alternative, related solution is a policy by which incomplete tasks are transferred to senior personnel after a certain time window. This "task transfer" option is explored in more detail in the next chapter. Both solutions are attractive in that they enable the enforcement of a strict FIFO workload policy while giving analysts a mechanism to disposition tasks that are legitimately stalled.

6.4 Summary

High WIP levels and a discretionary workload policy create a dangerous situation in which both managers and analysts will be tempted to prioritize tasks according to short-term objectives. The long-term process impacts are significant. Process variability increases, large numbers of difficult tasks are cached, and cycle times for difficult tasks explode. A FIFO workload policy effectively eliminates this potential.

7 Teamwork

The discussion to this point has focused primarily on the impact of individual analysts' workflow policies on process variability and efficiency. This chapter expands the scope of the investigation to study the performance of teams. In particular, we explore team structures as a means to boost productivity while simultaneously encouraging collaboration and knowledge sharing among analysts. The latter, team-oriented effects are a critical point of interest. Previous chapters have assumed that large performance discrepancies between analysts are an immutable artifact of data analysis production lines. This chapter challenges that assumption by seeking a team structure that reduces these discrepancies through improved collaboration and knowledge sharing.

The chapter begins by analyzing the skills-based team structure currently employed by Whitehead's Finishing Group. The structure groups analysts by their skill level, routing more difficult tasks to experienced analysts. The reader will recall from Chapter 5 that a team's productivity can be improved by this routing when advanced analysts possess a relative advantage in difficult tasks. Skills-based task assignment has other benefits: analysts manage tasks until their completion, giving them a sense of ownership; it also reduces the chances that complex tasks will get stuck with junior analysts who do not know how to handle them.[16] However, calling skills-based task assignment "teamwork" is somewhat of a misnomer. Analysts continue to work independently. Junior analysts' growth opportunities are reduced because they are no longer exposed to complex tasks. And, because analysts do not share tasks, the structure also does little to encourage collaboration and knowledge sharing. Thus, while improving short-term productivity, skills-based routing does not invest in teamwork or the long-term growth of team members.

Following the analysis of skills-based structures, an alternative concept, called the *triage model*, is considered. In this structure, tasks are distributed uniformly to analysts. However, junior analysts must transfer tasks to more senior analysts if they are unable to complete them within a certain amount of time. The triage approach has important training-related advantages over skills-based routing: junior analysts are repeatedly exposed to complex tasks, giving them the opportunity to learn the skills necessary for advancement; junior analysts also learn from the feedback they receive from senior analysts. The model, nonetheless, has complications: analysts must share credit for tasks. Like the skills-based model, there may also be a stigma for analysts labeled as beginner or intermediate.

[16] The models have not adequately represented the potential for tasks to become permanently lodged in analysts' queues as a result of their inability to find a solution. In practice, however, this happens quite frequently.

Complexities aside, the triage model is shown to be an attractive alternative to both random and skills-based task distribution models. Through discussion and simulation, this chapter shows that the triage model achieves some of the productivity gains of skills-based routing while simultaneously creating an environment poised for knowledge sharing and collaboration.

7.1 Simulation setup

In the simulations that follow, there are six analysts: two beginner, two intermediate, and two advanced analysts. Two types of tasks enter the workflow: captured and uncaptured gaps. An analyst's chance of completing the task is related to his or her skill level. Advanced analysts are assumed to have a strong relative advantage in uncaptured gaps over captured gaps, as indicated in Table 15. (These figures are slightly different from the baseline assumptions of Chapter 3.)

Table 15. Finisher performance assumptions by skill level.

	Beginner	*Intermediate*	*Advanced*
Gap closing probabilities			
Captured gap / cycle (%)	0.32	0.40	0.48
Uncaptured gap / cycle (%)	0.14	0.20	0.26
Average time / gap / cycle			
Captured gap (hrs)	1.20	1.00	0.80
Uncaptured gap (hrs)	1.95	1.50	1.05
Effective gap completion rate (based on above)			
Captured gap (gaps/hour)	0.27	0.40	0.60
Uncaptured gaps (gaps/hour)	0.07	0.13	0.25
Relative performance v. intermediate			
Captured gap	67%	100%	150%
Uncaptured gap	54%	100%	186%

7.2 Scenario 1: Skills-based Teaming

In skills-based team structures, analysts are organized into teams according to their estimated skill levels. As tasks enter the group's incoming queue, they are binned according to difficulty. Using linear optimization in Chapter 5, we showed that in the current system configuration it is optimal for advanced analysts to work on difficult tasks (i.e. uncaptured gaps.) Beginner and intermediate analysts, on the other hand, should work on easy tasks (i.e. captured gaps.) When advanced analysts have spare capacity, they too should contribute to work on easy tasks. Finally, we assume that, once routed to an analyst, a task is processed to completion.

7.2.1 The Weaknesses of Skills-based Teaming

While skill-based grouping yields improved productivity as shown in Chapter 5, it does little to encourage the collaboration and knowledge sharing reminiscent of true teams. First, analysts continue to function as independent contributors, owning tasks from start to finish. Moreover, while they are now teamed with analysts of similar skill, the structure per se does not encourage collaboration and exchange. In fact, skills-based grouping may reduce collaboration relative to the old team structure. Previously, beginner analysts were often grouped with advanced analysts and therefore likely to benefit from their insights and advice.

A skills-based team structure also precludes the exchange of ideas that inevitably occurs when tasks are shared. When analysts exchange tasks, different methodologies are vetted, creating a context in which the advantages of each method can be compared. If a task is passed from a beginner analyst to a senior analyst, the senior analyst may be able to offer comments and advice that will help improve the beginner's skills. If a task is passed from one analyst to another of similar skill level, the new owner may have an insight that the old owner did not consider. If, with its previous owner, a task had not meaningfully progressed for several cycles, then transferring it to another analyst might yield new insights, saving the group valuable work cycles.

Finally, and perhaps most importantly, skills-based task assignment eliminates a key learning opportunity for junior analysts and raises the specter of monotony for all. Certainly, specialization means that some analysts will be able to hone their productivity on a small range of tasks. However, analysts' exposure to the broad range of tasks entering the group is greatly reduced. Beginner and intermediate analysts see very few of the complex tasks assigned to advanced analysts, eliminating an important opportunity to learn through exposure. Likewise, advanced analysts no longer have the opportunity to work on simpler tasks; they may become frustrated with their consistently difficult workload.

Thus, while team-like in appearance, a skills-based grouping of analysts does little to encourage the type of knowledge sharing and collaboration one might expect from true teams. Analysts continue to function in relative isolation. Knowledge sharing and cooperation is illusive.

7.2.2 Implementation Complexities

Skills-based analyst grouping also faces a number of practical issues that make its implementation difficult. First, creating skills-based teams requires labeling analysts, a process that is bound to be imperfect and alienate some analysts. Analysts may resist being designated as "beginner" or "intermediate". Worse still, stigmatized analysts may reduce their effort in protest or out of alienation, making the designation self-fulfilling. A successful shift to skills-based teams requires strong leadership skills from management. Objective performance measures must form the basis for any skill-based grouping of analysts. More importantly, management must be able to explain the criteria by which analysts can advance into higher skill groups.

In addition to the management-related difficulties that arise out of placing analysts into skills-based groups, it proves difficult in practice to optimally route tasks. In Chapter 5, optimality was found by making assumptions about analyst skill level and task complexity. In practice, it is difficult to judge analyst skill and task complexity a priori. Determinations are usually made on empirical bases (i.e. by observing the analysts' performance over some period of time). However, empirical observations are problematic because analyst skill and task complexity are typically in flux. Analysts' skills improve with time. Likewise, task mix (i.e. the ratio of easy to difficult tasks) may change according to the state of the project, time of year, and customer needs.

The net result is that optimal task routing is a theoretically convenient but difficult-to-achieve goal. In its stead, managers must use assignment heuristics that strive for optimality while also meeting day-to-day process constraints. Hence, when an advanced analyst needs another task but there are no difficult tasks available, a manager should probably assign him or her an easy task, even though optimality suggests that advanced analysts should work only on difficult tasks. Similarly, because managers may not know the long-term task mix of their process, they should probably assign tasks in rough proportion to their arrival rate, even if that requires assigning tasks in a sub-optimal manner. For example, an influx of particularly difficult projects should be assigned broadly to all skill levels because the manager does not know how long the influx will last and cannot afford to underutilize the group's analysts.

Managerial and operational realities, therefore, make implementation of a skills-based team structure difficult. Analysts may resent the designation. Moreover, the optimality implied by carefully planned spreadsheets are often undone by operational realities.

7.2.3 Simulation Results

The diagram below illustrates the simple heuristic used to implement skills-based routing in simulation. New tasks are sorted into queues of uncaptured and captured gaps prior to assignment. In the simulation, 50 tasks are available for assignment at any point in time. An *Advanced Selector* selects tasks for the *Advanced Analysts* team from either queue; however, preference is always given to uncaptured gaps. Intermediate and beginner analysts receive only captured gaps.

This routing heuristic, while still somewhat stylized, should give the reader a sense for how optimal, skills-based routing would be implemented in practice. Unlike our linear optimization exercises of Chapter 5, which assumes we have a priori knowledge of all tasks, managers have only a small number of tasks available for assignment at any point in time.

Figure 26. Simulation of skills-based task routing.

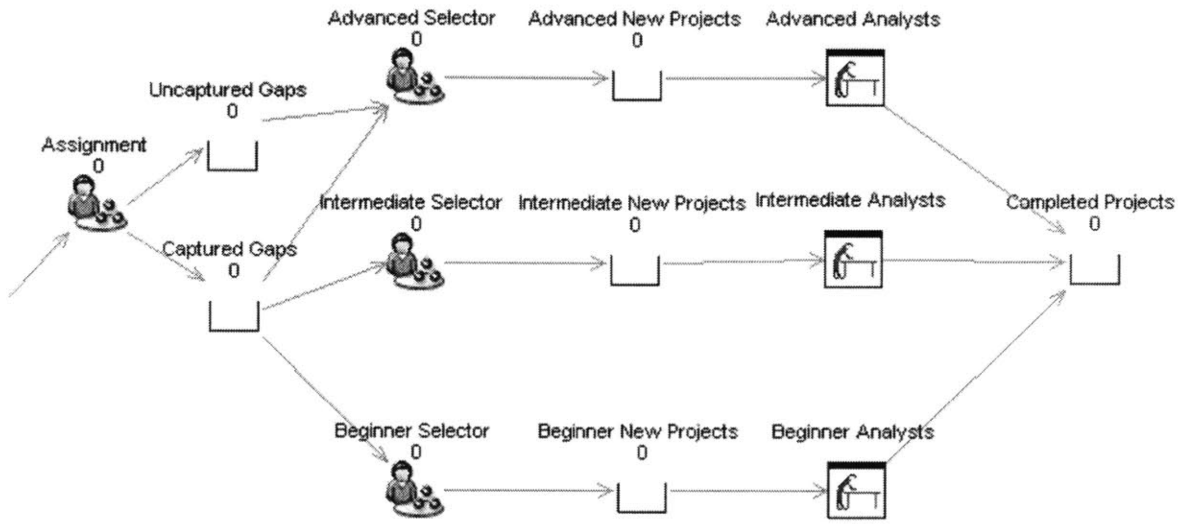


Table 16 shows the results of this skills-based routing heuristic. Results are contrasted against the optimal productivity predicted in Chapter 5 using linear optimization. They are contrasted against simulated results for random task distribution, as well as the predicted random distribution results from Chapter 5.

Table 16. Simulated results of skills-based task routing.

	Optimal distribution		*Random distribution*	
	Simulated	Predicted (Ch. 5)	Simulated	Predicted (Ch. 5)
Total tasks completed	7168	7240	6754	6808
Captured gaps (% total)	81%	81%	81%	81%
Uncaptured gaps (% total)	19%	19%	19%	19%
Performance v. random (predicted)	105%	106%	99%	100%

As predicted in Chapter 5, the simulations show that advanced analysts' relative performance advantage in uncaptured gaps can be translated into higher group productivity through skills-based task assignment. Moreover, our skills-based routing heuristic nearly matches the productivity predicted through linear optimization. The simulated system is simple enough that the heuristic is near optimal. In particular, by assigning from a large queue of 50 projects, we virtually guarantee that our optimality criteria can be met.[17] In a real operation, the heuristic might not

[17] Namely, the large queue ensures that there are easy tasks ready for assignment to beginner and intermediate analysts. This, in turn, permits us to assign difficult tasks exclusively to advanced analysts.

perform as well due to task mix fluctuations, unavailability of new projects for assignment, and other sources of noise not simulated in the above system.

7.2.4 Skills-based Teaming: Conclusions

On the surface, organizing analysts into teams according to their skill level is attractive: it boosts overall productivity and preserves analysts' sense of ownership over their tasks. Unfortunately, it also stigmatizes some analysts and is operationally difficult to implement. Most importantly, it does not encourage knowledge sharing and collaboration. Analysts continue to work as individual contributors, unaffected by the fact that their workgroup association has changed.

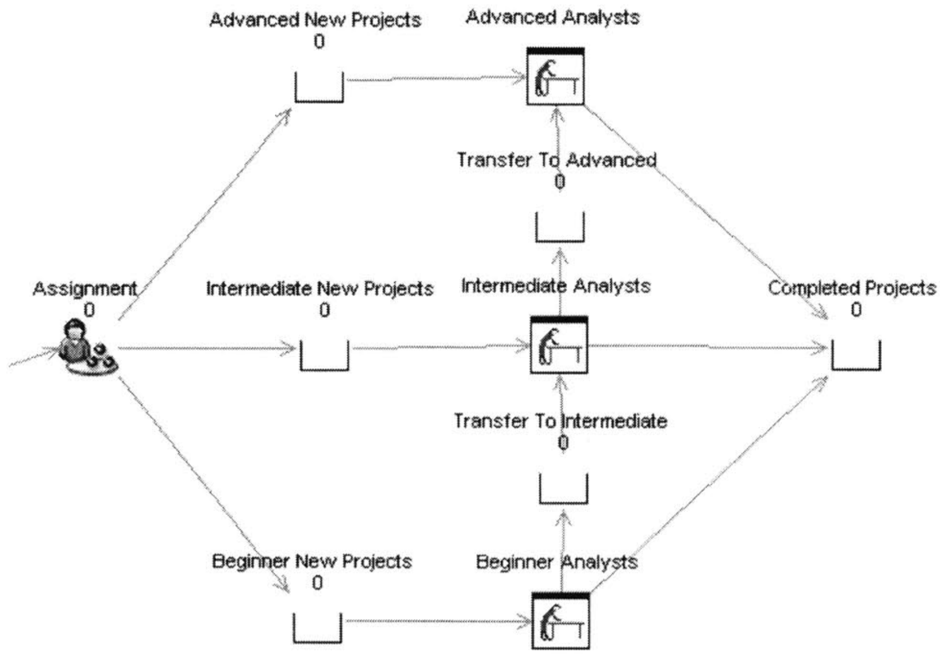
7.3 Scenario 2: Triage-model Teaming

The triage team model is an alternative solution that offers some of the advantages of skills-based task assignment while simultaneously encouraging knowledge sharing and collaboration. In the triage model, any analyst can be assigned any task. However, analysts that are unable to complete a project in some fixed number of cycles must pass it to the next skill level. In these simulations, a cutoff of two cycles was assumed. Correct setting of this value is a function of the number and capability of analysts at each skill level. This is discussed in detail in section 7.3.3.

When a task is transferred between teams, it is picked up at random by the next analyst in need of a new task. Intermediate and advanced analysts select their next task according to the following prioritization policy: they handle tasks returning from the lab first; next they look for projects being transferred from lesser-skilled teams; finally, when no other tasks are available, they select a new project.

The model is called a "triage" because junior analysts transfer projects that they have been unable to complete to more senior analysts. Figure 27 illustrates this team structure.

Figure 27. Triage Task Assignment Model.



In the simulations that follow, we assume that there is no value conveyed during the transfer of a project to a more senior person. More specifically, the fact that a junior analyst works on a task does not impact senior analysts' probability of completing the tasks. Thus, an advanced analyst closes an uncaptured gap with 26% probability on each cycle, regardless of whether the project was his or her own or transferred from another team. There are two contrary perspectives on this assumption: on the one hand, the junior analyst may rule out certain approaches to completing the task, saving the senior analyst time; on the other hand, the fact that the project was transferred may indicate that it is harder than originally estimated. In these simulations, we make the same simplifying assumption made throughout this thesis: that the probability of closing a gap in each cycle is independent of preceding cycles, and that it depends solely on the skill level of the analyst.

7.3.1 Standardization, Collaboration, and Knowledge Sharing

The triage model offers important advantages in both process control and team-orientation over skills-based grouping. First, merely because they may have to transfer a task, analysts will be driven towards standardized ways of describing tasks and their work. At Whitehead, project transfer is encumbered by the fact that analysts frequently use their own shorthand to describe work, making interpretation by other analysts difficult. With the prospect of tasktransfer, analysts will be driven formally (e.g. by published methods) and informally (e.g. by peer pressure) to describe their work in shared terminology. Once implemented, this standardization enables a higher degree of process monitoring and control than previously possible. Tasks can be inspected quickly to understand why they have not been completed within the expected number of work cycles. They can also be easily transferred if an analyst quits or takes a vacation.

By exposing all analysts to the full spectrum of tasks entering the group, the triage model also eliminates the boredom of specialization, while giving junior analysts important experiential learning opportunities. Though they may be unable to complete them, junior analysts are exposed to difficult tasks, without which they would find it difficult to learn the skills necessary to advance in the organization. Senior analysts benefit, too, through broad exposure. The routine of their job is reduced through exposure to easy and difficult tasks, lessening the chances that they grow bored or frustrated. Moreover, they continue to refresh their skills on less complex tasks, should they ever need to return to those tasks or offer advice to junior analysts.

True to its name, the triage model also reduces the chances of tasks becoming "stuck" in the workflow of analysts that are unable or unwilling to complete them. At Whitehead, particularly near the end of the HGP, there were many projects that had received eight or more cycles of work In most cases, the finisher had long since exhausted his or her "bag of tricks" and had begun using techniques that were inappropriate or unlikely to succeed. The triage model keeps tasks moving in both literal and psychological senses. Analysts that are unable to complete tasks are now able to transfer them to someone with higher skill and a fresh perspective. Psychologically, the triage model is also likely to create a sense of urgency: analysts know that they must complete a task in two cycles; failure means that their work becomes someone else's burden.

Lastly, and most importantly, the triage model establishes an organizational structure that encourages collaboration and knowledge transfer between analysts of different skill levels. If a junior analyst follows inappropriate methodology and, as a result, is unable to complete a task, he or she will no doubt receive a rebuke from the senior analyst who inherits the task. Conversely, junior analysts that have successfully boosted their skill level may be able to communicate new strategies to senior analysts; in turn, senior analysts are pressured to continue boosting their own skill level in order to solve tasks that their junior peers have been unable to complete. Additionally, because a junior analyst's task may be transferred to any analyst in a more senior team, the junior analyst also has the opportunity to collaborate with many people in the organization, gaining a variety of perspectives. Over time, the net effect of all of these communications is that senior analysts transfer knowledge and capability to junior analysts.

From process control and teamwork perspectives, the triage model offers many advantages over skills-based team structures.

7.3.2 Simulation Results

Despite the triage model's apparent advantages, they come at the cost of sub-optimal performance. Rather than routing difficult tasks (uncaptured gaps) immediately to senior analysts, as the optimality results of Chapter 5 might indicate, difficult tasks may also be routed to junior analysts. Only when junior analysts fail to complete the tasks are they then transferred to senior analysts. Because junior analysts possess a relative performance disadvantage in difficult tasks, the net effect is lower overall performance relative to the skills-based model.

Table 17 contrasts the simulated results of the triage model against simulated runs of skills-based and random task distribution. Results represent the average of 10 simulated trials for each team structure.

Table 17. Performance of triage-model teams versus skills-based teams.

	Triage Model	*Random*	*Optimal (skills-based)*
Total tasks completed	6862	6754	7168
Captured gaps (% total)	82%	82%	81%
Uncaptured gaps (% total)	19%	19%	19%
Performance v. Random	102%	100%	106%

The data shows that the triage model does not perform as well as the optimal, skills-based model, but slightly outperforms the random task distribution model. The gains are nonetheless fairly negligible (102% for triage v. 100% for random). As discovered in Chapter 5, the discrepancy between skills-based and random task distribution is most significant when senior analysts possess a strong *relative* advantage in one task over junior analysts. In these models, senior analysts are only slightly stronger in a relative sense when dealing with difficult tasks.

With such meager performance discrepancies, it is tempting to conclude that team structure and task routing are insignificant until one considers all of the factors not properly represented in the simulations above. First, the random model does not adequately represent the possibility that a portion of the difficult tasks will become *permanently stuck* with junior analysts who are incapable of completing them. This liability promises to lower the performance of a random model in practice. The triage model effectively eliminates the sticking problem. Second, as discussed earlier, there are operational constraints that preclude reaching optimal performance. These factors, such as short-term fluctuations in task difficulty, would reduce the performance of the optimal model but have no impact on the performance of the triage model, which is insensitive to task mix.

Most importantly, the simulated figures above cannot account for the long-term performance gains that come from the knowledge sharing enabled by the triage model. The random distribution model enables some degree of analyst training by ensuring that junior analysts get exposed to complex tasks; however, that exposure is not accompanied by the insights or oversight of more experienced analysts. In some respects, the skills-based model actually stifles learning by denying junior analysts exposure to more complex tasks. Only the triage model is able to achieve broad, institutionalized collaboration between analysts.

7.3.3 Load Balancing

The triage model has an additional attribute that makes it easy to manage: it is simple to balance the workload across analysts. Because senior analysts place a high priority on work being transferred from junior teams, they can automatically compensate for an influx of incomplete projects by not taking on new projects. Conversely, if the capacity of junior teams increases, senior teams can augment their reduced workload with new projects.

To see what percentage of tasks is transferred between teams, we calculate the probability that the tasks are not completed before the proposed cycle limit. Recall that in our model tasks are transferred after two cycles, and that there are two analysts in each skill group. Table 18 reflects the percentage of tasks started by each team that are either completed by that team or passed to a higher skill level team. For example, of the captured gaps started by beginner analysts, 54% are completed by the beginner analysts themselves, 30% are completed after being transferred to intermediate analysts, and 16% are completed after being transferred first to the intermediate analysts and then later to the advanced analysts.

Table 18. Task transfer rate in two-cycle triage model.

	Completes Task		
Starts Task	Beginner	Intermediate	Advanced
Captured Gaps			
Beginner	54%	30%	16%
Intermediate		64%	36%
Advanced			100%
Uncaptured Gaps			
Beginner	26%	27%	47%
Intermediate		36%	64%
Advanced			100%

To see how these transfers translate into time use, we need to first calculate the rate that tasks are transferred, and second, calculate the time required by the receiving team to complete the transferred tasks. Because the senior teams are more efficient than junior teams, they will generally require less time to complete the transferred tasks. For example, if the beginner team transfers 30% of its tasks to the intermediate team, those tasks will require less than 30% of the intermediate team's capacity. The exception, however, is the advanced team, which must finish tasks transferred to it. Thus, while beginner and intermediate analysts' contribution is limited to two cycles, advanced analysts may require more than two cycles to complete the task.

The table below summarizes the percent of capacity consumed by tasks transferred from junior teams. The calculations assume that the teams work on either captured or uncaptured gaps exclusively. Thus, if all teams processed nothing but captured gaps, advanced analysts would spend 14% of their time working on tasks transferred from beginner finishers; 24% of their time working on tasks from intermediate finishers; and 63% of their time working on new tasks of their own.

Table 19. Capacity utilized by transferred tasks in triage model.

	Completes Task		
Starts Task	Beginner	Intermediate	Advanced
Captured Gaps			
Beginner	100%	37%	14%
Intermediate		63%	24%
Advanced			63%
Uncaptured Gaps			
Beginner	100%	55%	53%
Intermediate		45%	43%
Advanced			4%

The situation for uncaptured gaps is somewhat different. The calculations indicate that advanced analysts spend most of their time processing tasks transferred from other teams, and very little time (4%) processing their own tasks. If the flow of uncaptured gaps from junior teams grew slightly, it would outstrip the advanced analysts' capacity to complete them. This data indicates that given the current system configuration, transferring uncaptured gaps after only two cycles may unbalance the system. To compensate, management might consider increasing the transfer threshold to three cycles or attempting to increase the number of advanced analysts on staff.

These figures indicate that some explicit management of the workload allocated to analysts is necessary. However, relative to skills-based grouping, little a priori knowledge of analysts' capacity is required. Management can simply observe whether transferred tasks are taking too much or too little of senior analysts' time and then adjust the transfer threshold accordingly.

7.3.4 Implementation Complexities

While offering many obvious advantages, the triage-model faces a number of implementation complexities. First, as in the skills-based model, analysts may still feel stigmatized from being placed in a beginner or intermediate skill group. The sting of this stigma is at least mollified by the prospect of learning the skills necessary to advance to the next skill level, something that would not be possible if their scope were limited to easy tasks as in the skills-based model. The stigma is nonetheless a liability and presents similar management and incentives problems as the skills-based model.

For managers, the triage model offers a unique means for measuring analyst performance. By looking at the percentage of tasks an analyst is able to complete without transferring to a more senior team, a manager can determine the analyst's efficiency relative to his or her peers. Thus, a beginner analyst that completes 60% of his tasks (transferring 40%) is performing significantly better than a peer that completes only 30% of his tasks. This *turnover* metric represents a rather precise and convenient means of measuring analyst performance. However, its use must be balanced against the pressure it places on the analyst: the analyst may be tempted to cut corners and force the completion of tasks in order to boost their turnover. A turnover metric, therefore, must be used in conjunction with other quality and performance metrics.

Finally, there is the question of how to assign credit for completed tasks to analysts. In scenarios where multiple analysts contribute to the completion of a task, do they share credit equally? Does the last analyst that worked on the task receive exclusive credit? Or, should the credit be weighted by seniority or time committed to the task? Each approach has advantages and disadvantages in terms of the incentives it creates. Regardless, success of the triage model clearly depends on a sense of camaraderie and teamwork between analysts. An incentive system that fails to engender this will render the triage model divisive. This is the topic of the next chapter.

7.3.5 Triage-model Teaming: Conclusions

The preceding discussion has highlighted the advantages of the triage model, while also exposing its fundamental tradeoff. In exchange for the training and collaboration opportunities that triaging enables, the system's overall performance is degraded slightly. The true nature of the tradeoff is difficult to explore since the advantages of better teamwork are prospective and not easily quantified. However, because optimality is difficult to achieve and comes at the cost of improved teamwork and collaboration, the tradeoff appears worthwhile. Institutionalizing the exchange of tasks and ideas between analysts is an important enabler of their continued growth. Ultimately, the triage model represents an investment in a group's long-term productivity; the same cannot be said of other team structures.

7.4 Summary

This chapter contrasted two team structures. *Skills-based teaming* promises near-optimal performance; however, it is difficult to implement and stifles analysts' learning and growth opportunities. The proposed *triage model* offers somewhat lower performance, but in the process enables a high degree of knowledge sharing and collaboration among analysts.

8 Organizational Study

Thus far, the thesis has focused on the impact that workflow policies have on process variability. In this chapter, we expand our scope to look at the impact that organizational dynamics have had on the process. In particular, we examine the reasons behind the significant performance discrepancies that persist between finishers. Some of these discrepancies can be attributed to differences in natural aptitude. This, however, is only part of the story: an organizational emphasis at Whitehead on individual contribution has undermined finishers' incentives to work as a team. Without incentives to collaborate and share their knowledge, finishers' differences remained insulated from change, leading to still wider performance discrepancies. These discrepancies, in turn, have made the process difficult to predict and control.

The chapter begins by setting the organizational context in which my internship began. Next, it analyzes the formal structure, politics, and culture of the finishing organization, showing that they are geared more towards individual contribution than teamwork. With these analyses as background, I then review several change initiatives I undertook during my internship and discuss why they met with only partial success. Finally, the chapter concludes with a recommendation for a new performance review and incentive system that clearly emphasizes and rewards teamwork.

8.1 Organizational Background

8.1.1 Finishing the Human Genome Project

My internship with the Finishing Group at Whitehead's Center for Genomic Research began under auspicious but tense circumstances. By March 2002, the group was just one year from finishing Whitehead's portion of the Human Genome Project (HGP.) The HGP represented the largest, most ambitious genomics effort ever undertaken. Whitehead had played a major role in the thirteen-year effort and looked forward to sharing in the celebration of its completion. At the same time, for a variety of reasons, Whitehead found itself at risk for not finishing on time. Because funding for future projects hinged on proving its capabilities in the HGP, senior management at Whitehead applied significant pressure to accelerate the project. As the final stage in the gene sequencing process, the Finishing Group inevitably bore a great deal of this pressure.

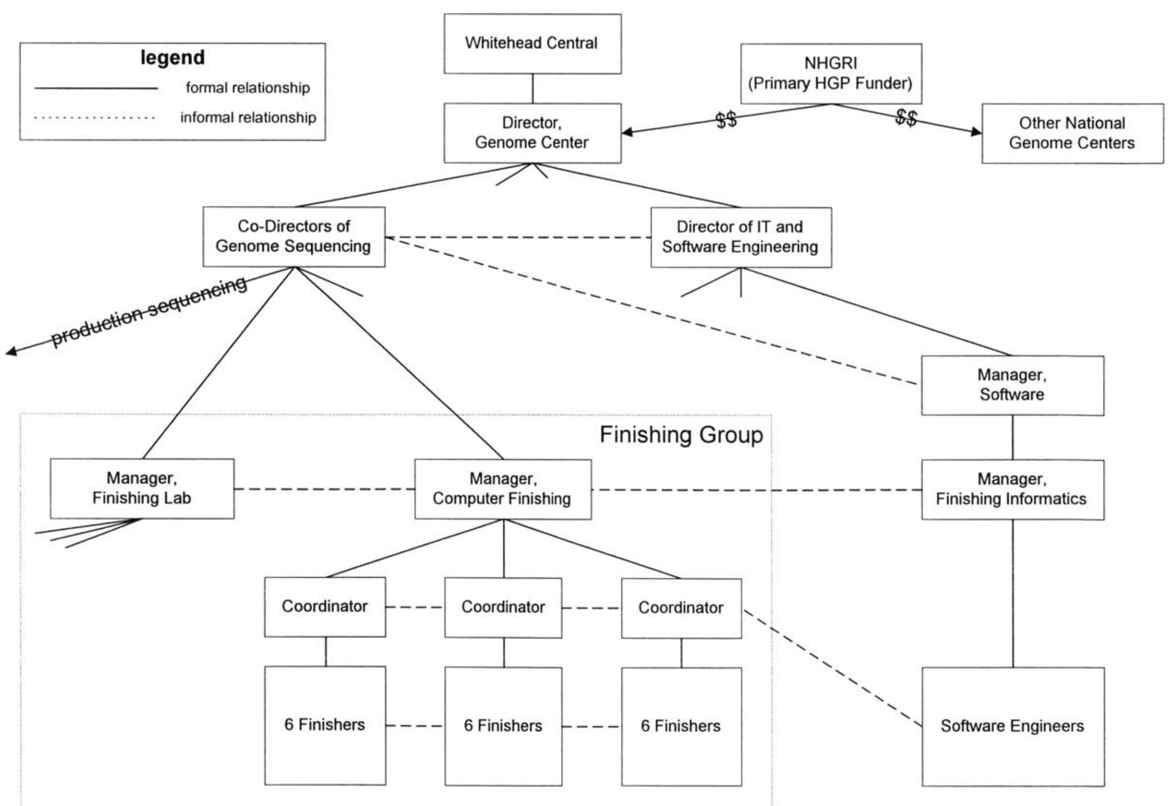
8.1.2 Organizational Structure

Whitehead's Computer Finishing group is structured in a hierarchical fashion typical of many production-oriented organizations. At the time my internship began, the group was organized into three groups of approximately six finishers each. Each group of finishers sat together in a bullpen office and reported to a coordinator. The three coordinators, in turn, reported to the manager of the Computer Finishing Group.

Computer Finishing is supported by two other organizations: the Finishing Lab is responsible for executing laboratory procedures ordered by the finishers; and, a group called Finishing Informatics is responsible for developing database and software tools used by the finishers. The managers of Computer Finishing, the Finishing Lab, and Finishing Informatics report to the Co-Directors of Genome Sequencing. The two directors are jointly responsible for all of Whitehead's gene sequencing operations, including finishing. The most significant project underway at the center, in terms of both revenues and staffing, was the Human Genome Project.

These directors, in turn report to the Director of the Genome Center, who holds both production and research responsibilities at Whitehead. Outside the Genome Sequencing group, the Genome Center primarily conducts long-term research into fields such as cancer and comparative genomics. In fact, with the exception of the production-oriented Genome Sequencing group, Whitehead is primarily a research institution like its parent organization, MIT. Figure 28 reflects the organization as described. Solid lines reflect formal reporting relationships. Dotted lines reflect informal relationships and are a subject of discussion in the next section.

Figure 28. Organizational chart for Whitehead's gene sequencing group.



8.2 *Organizational Analysis*

In this section, we look at the finishing organization's structure, politics, and culture in order to understand why teamwork has been difficult to achieve. Together, these perspectives paint the picture of an organization that holds individual achievement high, but fails to adequately recognize the importance of teamwork.

8.2.1 Formal Structure

Whitehead's Finishing Group has a hierarchical, control-oriented structure that characterizes many production organizations. Hierarchy can be a successful approach to running a complicated operation; at the same time, successful hierarchical organizations depend on informal networks and strong management to bridge otherwise rigid organizational boundaries. Unfortunately, because it was a young organization that grew quickly, the Finishing Group lacked many of these important organizational capabilities. As a result, thegroup tended to lack a sense of teamwork and collaboration.

In early 2002, the Finishing Group tripled in size as management increased its staffing to meet the pressures of the HGP. The growth occurred so rapidly, however, that many finishers failed to develop informal relationships with other members of the gene sequencing operation. Twenty finishers sat in three neighboring rooms, but often knew only the people in their own room. Even within a room, the strength of finishers' relationships varied greatly. One room of finishers grew quite fraternal, while the other two rooms tended to be quiet and formal. This sense of isolation extended beyond the finishing group. Few finishers had relationships with the Finishing Lab or Finishing Informatics, despite the fact that the groups worked together closely. In short, the organization had grown ahead of its members' abilities to develop robust personal networks.

Leadership problems tended to aggravate finishers' sense of isolation. Coordinators were saddled with large workloads that limited their ability to train their finishers or facilitate team-related activities. As recently promoted finishers, the coordinators also had limited experience in leading large teams. As a result, a finisher's placement on a team tended to have limited direct consequence on their daily affairs. Team meetings were rare, as were individual meetings between finishers and their coordinators. Finishers tended to work in isolation. What little collaboration did occur was the result of the few personal relationships that they managed to develop.

In summary, Finishing grew into a large hierarchical organization, but unfortunately lacked many of the capabilities important for success. Limited informal networks left finishers feeling isolated from their peers. And, coordinator groups tended to be teams in name only.

8.2.2 Political perspective

Two themes dominated the political landscape at Whitehead's Genome Center in late 2002. First, there was the pressure to complete the Human Genome Project by April 2003. This pressure started at the top of the organization and radiated downward in a manner that sometimes harmed long-term productivity and group morale. A second important political theme was thedifference in career objectives that existed between the Center's management and its finishers. This gap created a level of distrust between the two parties. It also hampered management's ability to recognize the morale and incentive problems stirring in the finishing organization. Together, these political trends made teamwork and collaboration a difficult proposition.

By the summer of 2002, the pressure to complete the HGP was palpable. Work upstream of finishing had begun to dry up, shifting focus to the finishing group, which held the last cluster of work to be completed. For a variety of reasons, Whitehead found its finishing efforts behind those of other centers. As 2002 progressed, enormous pressure was applied to the finishing organization to accelerate its progress. At a loss for how to achieve this acceleration, management pursued a number of short-term initiatives to boost output. Many of these initiatives did not entail true gains in operational efficiency; rather, difficult work was simply deferred. The initiatives took a heavy toll on finishers, who found them disruptive and inefficient. Moreover, as the year progressed, the difficult work that had been deferred came back to haunt the finishers. Remaining work became increasingly difficult, but the pressure to boost output continued unabated. With little guidance on how to achieve these gains, finishers became increasingly frustrated and demoralized.

These pressures were aggravated by the differences between finishers and senior management. The Center's senior management consisted primarily of academics and researchers, many of whom had pioneered modern gene sequencing. Typically Ph.D.'s, these managers held long-term career interests in the field of genomics. Though they ran a production facility, their ambitions also included publication, tenure, and senior roles in industry. This background contrasted sharply with that of the finishing personnel. Coordinators and finishers were typically young, in possession of a bachelor's degree, and at an early stage in their career. Some aspired to long-term careers in genomics or medicine. For others, finishing represented a temporary stopping point on the way to other careers. Unlike senior management's leadership roles, a finisher's job had few obvious growth opportunities; Whitehead was a research organization with limited long-term potential for someone in a production role. Thus, although they worked closely together, finishers and management approached their work from very different circumstances.

What tied these disparate groups together – and what should have formed the basis for bridging the gaps between them – was the sense that everyone was contributing to something of great importance. The Center's management and finishers shared a common inspiration in their work on the HGP: to perform the genomics equivalent of landing a man on the moon. Finishers, by virtue of their rank in the organization, did not stand to capitalize on the success of the HGP in the same manner as senior management. Nonetheless, they were sufficiently inspired by the endeavor that simple recognition of their contribution represented a significant incentive. This was particularly valuable for an institution that had little leeway in its financial incentives because it was publicly funded.

The Center's management, however, failed to take advantage of these opportunities. In one instance, a coordinator and several senior finishers complained that their names had not been added to the authors list associated with DNA submissions to NHGRI, the government body coordinating the Human Genome Project. The situation was eventually righted, but it took many months. There were other lost opportunities. A visit to another major sequencing center, Washington University, revealed that they used a variety of innovative, non-financial incentives: high-performing finishers might be recognized in front of their peers at a monthly meeting or receive a signed letter from Francis Collins, Director of NHGRI. Other genome centers established standing policies whereby accomplished finishers were sent to conferences, talks, or other sequencing centers.[18]

In summary, though the finishing organization was under significant pressure, there remained important opportunities to incentivize finisher productivity and boost morale. The Center's management did not act, perhaps because it was unable to sense just how frustrated finishers truly were. Inaction, however, communicated the message that finishers' contributions weren't important. In several discussions, the manager of the finishing process intimated that these initiatives would be taken up once the HGP was complete. Yet these tense circumstances were precisely the time at which such incentives were so important.

8.2.3 Cultural Perspective

Whitehead's history as an innovative research center created culture that prides itself on individual achievement and new technology development. These capabilities helped propel Whitehead to the forefront of gene sequencing. At the same time, however, Whitehead's continued emphasis on individual contribution and technology tended to overshadow teamwork, coordination, and other capabilities that became increasingly important as the organization grew.

Like Whitehead at large, Finishing's emphasis on individual contribution grew out of its history and the complex nature of its work. When the Finishing Group was created, finishing was essentially a research process. A small group of finishers developed many of the methods used by

[18] Source: Jane Wilkinson, on staff at Whitehead, previously led a finishing group at the Sanger Institute.

today's finishers. Because the team remained a small group for several years, its self-image tended to reflect the dedication and individuality of its members. This selfimage was still evident when I joined the Finishing Group: the coordinators and several senior finishers continued to work evening and weekend hours, particularly near the end of the HGP. While this dedication proved invaluable to the group's productivity, it set a high bar for the fourteen finishers that joined the organization in 2002. It also meant that when pressure was applied to boost output, the group tended to rely on the heroic acts of a few members rather than invest in the long-term productive capacity of some of its slower finishers.

Finishing's cultural emphasis on individual contribution is also an outgrowth of the work itself. Finishers perform complex tasks that often require hours of sustained concentration. Naturally, they develop a strong sense of pride and ownership for their work. Pride in their work is critical because, without it, finishers might easily become frustrated and fail to complete some of their more difficult tasks. At the same time, the sense of ownership that finishers bring to their work heightens the perceived value of individual contribution. Team-oriented activities become counter-culture. Consider the example of two finishers sharing a task: neither finisher would feel a strong sense of ownership for the task; there might also be concerns about who would receive credit for the task's completion. Finishing's complex nature therefore proves to be a double-edged sword: it inspires a strong sense of ownership, but it also makes teamwork more difficult.

A history of innovative technology development may also have made it difficult for the Finishing Group to recognize when a problem necessitated better teamwork, rather than a technological solution. One such example arose late in the HGP. Failed coordination between finishers had led to a situation in which redundant portions of the genome were being processed. To accelerate the HGP, it became critical to eliminate this redundancy. Almost instinctively, several managers proposed using a spreadsheet to collect redundancy information. Unfortunately, technical solutions such as these were part of the problem: informatics-based tools had insulated finishers from the need to communicate and collaborate.

As an alternative to the spreadsheet-based approach, I proposed that we use a rudimentary, paper-based system in which finishers would post their work on a large shared wall. To determine if they were working on a redundant part of the genome, finishers had to leave their offices, examine the work of their peers, and then jointly strategize a solution. The process proved considerably messier than an informatics-based solution, but was an instant organizational success. For three days, finishers were out of their office, communicating and collaborating more than I had seen them do in the preceding six months. The project eventually came to an end and finishers returned to their old norms of largely independent work. For a short while, however, the importance of old-fashioned, face-to-face communication was vetted for everyone to see.

In summary, though an emphasis on individual contribution and innovation played a critical role in Finishing's success, this emphasis also limited the organization's ability to recognize the importance of teamwork and other organizational capabilities.

8.2.4 Analysis: Summary

By early 2002, the Finishing Group had reached a size and complexity that made teamwork critical. The organization was simply too large for its members to learn without the aid of better communications and collaboration. In many ways, however, the organization's structure, politics, and culture remained focused on individual achievement.

8.3 Change Initiatives Undertaken

This section describes several change initiatives I undertook during my six-month internship with the Finishing Group. Each of these initiatives met with moderate, but not unqualified, success. Two initiatives, in particular, highlight important characteristics of the Finishing Group, while pointing to future areas in need of change: the rollout of a new performance measurement system and the implementation of a new peer review process.

8.3.1 New metrics for finisher performance

Historically, management has measured productivity by counting the number of projects a finisher completes. Unfortunately, the method is not entirely fair and is also susceptible to manipulation. First, it does not account for project complexity, which may vary significantly. Second, coordinators control project assignments and may be tempted to preferentially assign easy projects to their own teams. Finally, because the group lacked strict project assignment protocols, finishers often accumulated large queues of work, making it easier to "skim" easy projects off the top of their workload and temporarily boost output. In short, a project-based measure of finisher output provides a partial, and often biased, view of their true productivity.

To counter this inequity, in September 2003, I proposed a new project-rating scheme that I called the *Difficulty Index*. In the proposed scheme, individual projects would be scored according to their estimated difficulty, with complex projects scoring higher than easy projects. By summing the scores of the projects a finisher completed over the course of a year, a manager could determine how productive that finisher was relative to his or her peers. Because the system might still be prone to manipulation, I further proposed that projects be assigned randomly without coordinator intervention.

The response to my proposal was mixed. On the one hand, it was very hard to deploy a new performance measurement system amidst the stresses of the HGP. On the other hand, the organizational support I had expected was missing. Finishers, despite having clamored for an objective system, greeted it with skepticism, suspicious that it represented yet another vehicle for management control. The three coordinators in the group felt that the system was too simplistic. The only significant support the new system received came from senior management. They wanted to deploy it but were wary of doing so given the late stage of the HGP.

In fact, the proposal lay dormant until the end of 2002, when it came time to do performance reviews. These were awkward circumstances under which to introduce a new measurement system. First, finishers had not yet had the system adequately explained to them. Second, the original proposal called for new project assignment policies, most of which had not yet been implemented. In essence, the system was deployed at a time that was convenient for management. An important opportunity to hold a frank, open discussion about the merits of the new system with finishers had been missed.

8.3.2 Peer review

In December 2002, working with finishers and managers, I sketched out a finisher peer review process. In the proposed process, finishers would meet regularly to review each other's projects and offer recommendations. The plan seemed like a novel solution to the organization's training needs. It would provide finishers with the opportunity to discuss problems without fear of being judged by their coordinator. It would also give the group a natural means to identify best practices from among the many methods employed in finishing. Finally, we thought peer reviews would improve the level of communication and collaboration in the group.

For the process to be a success, I knew it would be critical to gain finisher support. Meeting individually and in small groups, I discussed implementation details with the finishers. In these meetings, finishers made suggestions about the size of the peer review committees, the frequency of the meetings, and the process for reviewing projects. I also let it be known that finishing management was in full support of the process, since many finishers were concerned that management would not condone this use of time. I contributed only two constraints to the implementation: everyone had to participate or else the process would fall apart; and, the committees needed to note their recommendations so that they could eventually be compiled into a list of best practices. Though there remained some open debate about specific details, a majority of the finishers seemed to favor implementing a peer review process.

One week before the end of my internship, a group of seven finishers held Whitehead's first finisher peer review meeting, while I joined as an onlooker. By all measures, the meeting was a success. Everyone participated and a large number of attendees told me later that they learned something valuable from their peers. I left my internship confident that we had installed a process that would improve knowledge sharing and collaboration between finishers.

Three weeks later, however, I learned that the meetings had fallen apart. Expert finishers complained that they spent most of their time teaching – but learned little. Others complained about the meetings' time requirements, suggesting that they be shorter, less frequent, and optional. As attendance became increasingly spotty, management responded by making peer reviews mandatory, a move which had the unfortunate effect of increasing finishers' distaste for the process. The peer review process had failed. In retrospect, the reason seems clear: finishers had very little incentive to think and act as a team.

8.4 Recommended future organizational changes

Ideally, in their efforts to embrace teamwork, Whitehead's managers would address the structural, political, and cultural impediments to teamwork simultaneously. Unfortunately, an organization's structure is usually the only aspect that can be changed on command. The political and cultural climate of an organization inevitably evolves more slowly and is difficult to manage explicitly. Nonetheless, revamping an organization's structure is a powerful change agent: done correctly, it can set the stage for positive changes in the political and cultural climate.

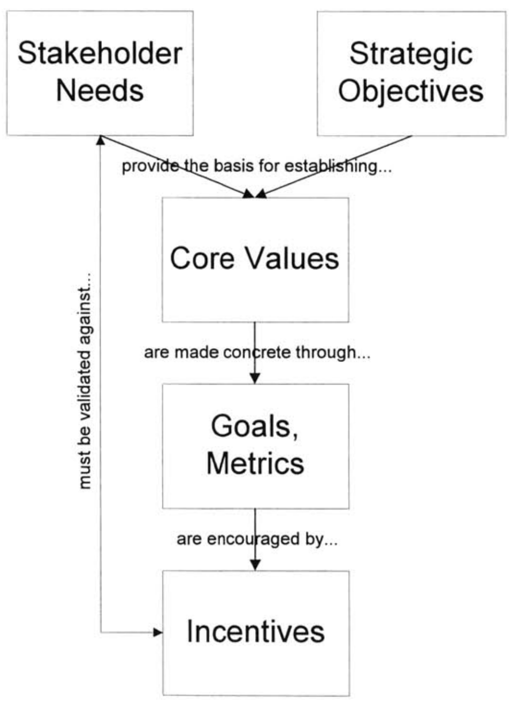
In this section, I outline a proposed revisal to the performance review and incentive system in Whitehead's Finishing Group. The proposal is based on the belief that team-oriented behavior will emerge if it is measured and rewarded. First, I provide a framework for thinking about the relationship between Whitehead's core values, its performance review process, and its incentive system. Next, I describe a simple performance review template that clearly links Whitehead's core values to the performance characteristics it values in its employees. Finally, I conclude by recommending changes to the organization's incentive system.

8.4.1 Review and incentive system should flow from core values

For an organization's performance review system to be successful, its stakeholders must share a common view of their organization's core values. This common understanding is often assumed to exist, but does not. Even when employees comprehend their organization's core values, they may fail to see how those values translate into personal performance goals. At Whitehead, finishers certainly recognize the importance of high productivity. They can see its importance in the way they are measured and rewarded. However, other core values are probably less obvious. Would finishers have been able to explain how their last performance review exemplified Whitehead's belief in innovation and employee growth? Probably not. Such disconnects create a major incentive problem: when employees do not understand their company's core values, or the link between those values and their own performance, they inevitably behave in ways that undermine the organization's objectives.

Understanding core values and how they relate to employee incentives is the focus of this section. Figure 29 illustrates the conceptual framework I used for thinking about Whitehead's core values and the link between those values and employee performance. In this framework, core values derive from a compromise between the organization's strategic objectives and the needs of its key stakeholders. Once core values are understood, managers can set specific goals and metrics to measure progress. Finally, incentives encourage performance towards those goals.

Figure 29. Linking core values to incentives.



Consider, first, the assumption that core values flow from both stakeholder needs and strategic objectives. The connection between a company's core values and its strategic objectives is obvious. Whitehead participates in a competitive industry. Dozens of commercial and educational institutions are in constant pursuit of technologies that will permit faster and faster gene sequencing. Whitehead's ability to stay competitive through technology and process innovation is critical to its ability to secure funding for future projects like the HGP. The strategic imperative of innovation is clear in Whitehead's case. Innovation is a core value.

The connection between stakeholder interests and an organization's core values is less obvious but equally important. Organizations, particularly those centered on human, analytically intensive processes like finishing, depend on the productivity and well being of their employees. Understanding employees' needs – career, personal, and otherwise – is critical to defining an organization's core values. The more an organization's core values reflect the needs of its stakeholders, the easier it will be for that organization to inspire its people to help meet its strategic objectives. Doing so, however, may put stakeholder needs at odds with strategic objectives. Therefore, determination of an organization's core values inevitably involves compromise.

When considering stakeholders' needs, it is important to recognize that some needs are shared while others are unique. For example, teamwork serves all stakeholders' needs by helping to eliminate boundaries between individuals and fostering a sense of cooperation. Some stakeholder needs, however, are quite different. Reconciling these different needs falls to the proper definition of an organization's core values. Finishers, by virtue of their background and role, have different objectives than their managers. They seek a learning environment and growth opportunities that will help jump-start their careers. These needs are different from those of Whitehead's managers, for whom research and public recognition are more important than basic learning opportunities. Whitehead's core values must embrace finishers' need for personal growth in the same way that it embraces managers' desire to forward their research.

In many ways, then, reconciling strategic objectives and stakeholder needs falls to proper definition of an organization's core values. For the Finishing Group, as with any organization, this reconciliation process is complex. Nonetheless, the preceding discussions provide insight into what those core values may be:

- **Individual Performance** – As a production organization that depends on the skills and dedication of its analysts, Whitehead must continue to recognize and reward individual performance. The organization has a history of emphasizing this value, but nonetheless must strive to define clear, objective measures of what constitutes exceptional performance. The Difficulty Index is one example system by which this can be accomplished. Nonetheless, the organization must also recognize the limitations of simple metrics. Quality of work, leadership, flexibility, and a variety of other factors must also be considered when measuring a finisher's individual performance.

- **Employee Growth** – This chapter has highlighted the sense among finishers that their efforts are inadequately recognized and that their growth opportunities in the organization are limited. This perception was enhanced by short-lived circumstances: the pressures of the HGP, leadership issues, and differences between finishers and management. Finishers' frustration, however, points to an important weakness in the current incentive system. Whitehead's core values, and the review process and incentive system that flows from it, must place renewed emphasis on the growth of its employees. Employees should be incentivized to seek out new learning and career opportunities within the organization.

- **Teamwork** – Collaboration, communication, and knowledge sharing are all critically lacking in the finishing organization. Some of this stems from temporary circumstances, but much of it owes to an historic emphasis on individual contribution. Finishers refuse to participate in team-oriented processes like peer reviews because the link between that participation and their own incentives is tenuous. Whitehead must establish teamwork as a core value and then seek out goals, metrics, and incentives that encourage it. Teamwork promises a more cohesive, collaborative work environment. More importantly, it promises to improve productivity and eliminate the glaring performance discrepancies that exist between the finishers.

- **Innovation** – The long-term competitiveness of Whitehead's Genome Center depends on its ability to continue innovating. Because Finishing continues to be a manual process, it has become increasingly expensive relative to the automated gene sequencing process. Finishers are in an excellent position to identify new techniques to improve the efficiency of their process. Yet, because the current incentive system fails to adequately reward this core value, many finishers have failed to invest time into the skills that might enable them to innovate. Several finishers have already ventured into software development; some have worked with the Finishing Lab to prototype new laboratory procedures. Finishers should be encouraged to channel some of their energies into new process development, documentation, data analysis, and other fruitful areas for innovation.

8.4.2 Alignment of current review system and core values

The current performance review process at Whitehead embodies the core values just described but possesses a number of weaknesses. Table 20 lists the characteristics (called *performance factors*) used to rate Whitehead employee performance on a yearly basis.[19] Factors in the left column reflect characteristics by which all employees are judged; factors in the right column are used primarily for managers.

Table 20. Performance factors used to measure Whitehead employees.

All employees	*Supervisors only*
Job knowledge & skills	Leading & influencing
Quality of work	Coaching, training, & mentoring
Commitment & dedication	Planning & goal-setting
Initiative	Judgment & decision-making
Meeting set objectives/tasks	Performance management
Dependability	Budget & financial operations
Planning & organizational skills	
Communication	
Critical thinking & problem-solving	
Collaboration & cooperation	

Managers are asked to rate their employees for each of the factors on a scale that includes *unsatisfactory*, *acceptable*, *commendable*, and *exceptional (UACE)*. The criteria for these ratings are specified in Table 21. Employees' overall performance is then determined by averaging their performance across all of the factors.

Table 21. Scale for Whitehead performance factors.

Rating	*Description*
Exceptional (E)	Performance clearly and consistently exceeds significant and demanding position requirements and expectations. The employee's results frequently exceed objective(s). This employee is easily recognized as exceptional among his/her colleagues in particular factor areas and/or in overall performance. There are no significant performance deficiencies. The quality and quantity of work consistently exceed that expected of someone fully qualified.
Commendable (C)	Performance consistently meets expected requirements and also occasionally exceeds expectations in significant areas. Capably handles all assignments, requiring only normal supervisory guidance.
Acceptable (A)	Key responsibilities and assignments performed reasonably well, but occasionally does not fully meet expectations and/or job requirements. Closer supervision and guidance usually required either in general or in particular job areas that need specific development or improvement.
Unsatisfactory (U)	Does not sufficiently meet several significant job requirements and/or frequently falls below standard performance expectations. Employee should be put on a performance improvement plan if not already on one, and must improve performance to retain employment.

On first glance, each of the four core values outlined above can be found in one or more performance factors. The factors also highlight a broad range of desirable features in an employee. Moreover, the system appears to strike a balance between objectivity (i.e. clear factors, clear definition of ratings) and subjectivity (i.e. manager has leeway to call out exceptional performance.)

Closer consideration, however, reveals a number of potential pitfalls. First, several of the factors overlap in meaning. *Commitment & dedication* seems a close cousin to *Dependability*;

[19] Courtesy of Nicole Barna, HR Director at the Genome Center

likewise, *Planning & goal-setting* is indistinguishable from *Planning & organizational skills*. Second, the factors appear to be of equal importance. Simply averaging the factors to produce an overall rating overlooks the fact that the relative importance of factors will vary by group, employee, and current work circumstances. Third, it is not clear which metrics should be used in determining the UACE rating. The qualitative bases shown in Table 21 do not tell a manager how to determine whether an employee "consistently exceeds" or "falls below" expectations. The ratings system also fails to capture the fact that some performance factors, like a finisher's productivity, actually have quantitative measures.

Perhaps the weakest aspect of the current review system is that it does not clearly link an employee's performance back to the organization's core values. While the performance factors hint as to what core values may exist in the organization, the link is tenuous and surely subject to interpretation. Moreover, teamwork, which has been a frequent focus of this thesis, is largely absent from the "all employees" set of performance factors. *Collaboration & cooperation* is the only factor used for a finisher. Other seemingly relevant characteristics like *Leading & influencing* and *Coaching, training & mentoring* are reserved for supervisors.

In summary, the current review system is broad in its coverage but weak in its link to core values and its commitment to specific goals and metrics.

8.4.3 Proposed Performance Review System

Table 22 reflects a proposed alternative to the performance factors currently used for finishers. It reflects a simpler, more direct approach to the review process. There are fewer performance factors and each is clearly linked to one of the four core values.

Table 22. Proposed performance review system.

Core Value	*Performance Factor*	*Description*	*Example*
Individual Performance	Productivity	Raw productive output, weighted by average difficulty	Number of projects submitted last quarter
	Quality	Quality of work completed	Number of projects rejected because they failed to meet quality standards
	Non-production activity	Work that did not directly contribute to production	Assisted efforts to expedite a special project outside normal production
Individual Growth	Performance improvement	Improvement in any of the performance factors listed above over previous performance periods	Reduced quality problems in project submissions over last quarter by 25%
	Training completed	Formal training completed; may include qualifying exams proving capacity in certain tasks	Worked in lab for a day to learn new process; completed a training seminar on programming/scripting
Teamwork	Standardization	Efforts undertaken to standardize the group's approach to common tasks	Documenting current best practices, methods, tools so others can use them
	Knowledge sharing	Efforts to distribute knowledge throughout organization; may be mentor or simply facilitator	Participated in peer review process; did a case review with peers; spent time mentoring peer
	Cross-functional work	Efforts to improve communications and coordination with external groups	Worked closely with lab to debug new laboratory procedure
	Team leadership	Leadership of peers by influence or example	Organized or facilitated peer review, training seminar
Innovation	New tools, methods	Development of new techniques or tools that improve the quality or efficiency of group	Wrote a script that automated a common task
	New initiatives	Assistance in identifying, exploring, or validating new techniques	Volunteering to work on an experimental project

8.4.4 Measuring Performance

By itself, the proposed performance factors are insufficient to encourage the type of team-oriented behavior desired from finishers. These factors must also be accompanied by clear metrics, objectives, and incentives. Table 23 illustrates a number of techniques that Whitehead should consider for measuring finishers' performance in the factors above.

Table 23. Techniques for measuring performance.

Metric	*Description*	*Where to apply it*
Scored project output	Direct measurement of productive output, weighted by difficulty.	Performance, Growth
360° peer review	Review from immediate peers commenting on employee's mentorship, communications, and teamwork.	Teamwork, Innovation
Customer/supplier review	Review from members of other groups that supply or receive work from employee, commenting on his or her quality of work, responsiveness, and communications.	Teamwork
Subordinate review	Review from direct report commenting on manager's leadership, communications, and organization.	Performance, Teamwork
Manager review	Review of employee from manager commenting on all aspects of employee performance.	Performance, Growth, Teamwork, Innovation

All but one of the measurement techniques listed above, *scored project output*, are qualitative. In fact, several of these metrics are already in use at Whitehead. The critical problem with the current system is not that its performance metrics are qualitative or subjective in nature. Rather, the problem is that these feedback collection mechanisms are not clearly linked to the performance factors. Thus, when a manager reviews an employee's *Communication* performance factor, it is not clear how much of that feedback is the result of his or her own assessment or the feedback of others. Giving managers a set of measurement tools and clear guidelines on where to apply them is critical. Finishers must know that there are multiple constituencies, besides their manager, that factor into their performance review.

8.4.5 Setting Objectives

Even with clear performance metrics, employees need to know how high to aim in their performance. Whitehead's finishers have persevered without performance objectives – an ambiguous situation that has heightened the anxiety associated with performance reviews and ensured that some finishers do not aim as high as possible. In addition to a new performance review process, Finishing should consider a quarterly or semi-annual process in which performance objectives are set and reviewed for each finisher. In the proposed process, the employee would meet with his or her manager to review the previous period's objectives and discuss how well they were met. The meeting would conclude with a new set of objectives designed to address performance shortfalls and any new strategic needs that may have arisen for the organization.

Establishing an objective-setting process is essential not only to guiding employee expectations but also to establishing a framework for continued dialogue about performance between the manager and the employee. Setting quantifiable, objective performance goals is bound to be a difficult and imperfect process, even in a quantitative, production-oriented environment. However, it is less critical for a manager and the employee to scrutinize individual goals than it is for them to meet regularly, discuss basic objectives, and align expectations. The meetings will provide a forum for the manager and employee to express performance-related concerns. More importantly, the meetings will give the employee an opportunity to correct performance issues long before a year-end performance review, when the stakes are high and it is arguably too late.

8.4.6 Incentivizing Performance

Whitehead's ability to incentivize its finishers is constrained not just by the fact that the Genome Center is publicly funded but also because of the HR practices of its parent organization, the Whitehead Institute. Specifically, strong financial incentives are often cited as being difficult, if not impossible, to arrange. With regards to these apparent constraints, two perspectives are worth considering. First, as mentioned earlier in this chapter, there are several non-financial incentives that have not yet been fully utilized. Recognizing high performance through public acknowledgement, trips to conferences, time off, and signed letters are all practices employed to some success by other genome centers. Finishers may also respond positively to title changes and additional responsibility, whether or not those are accompanied by financial compensation.

While non-financial incentives are viable, the Genome Center must also consider breaking with Whitehead tradition and developing a compelling set of production-oriented financial incentives for its finishers. Doing so has been difficult because portions of the Institute continue to labor under the belief that it is possible to construct an incentive system that can simultaneously serve the needs of both its research- and production-oriented organizations. Finishers are production personnel who should be rewarded on the basis of their productivity. Their goals are inherently more short-term and precise than those of professional researchers. For researchers, there is always the implicit incentive of publication, tenure, and fame to compensate for below-industry salaries. Finishers and other production personnel lack these same incentives, making financial compensation more critical.

There are two other reasons why Whitehead should consider more active use of financial incentives. First, as long as finishers' salaries remain low, Whitehead risks significant attrition. The financial costs of this attrition, given the slow learning curve of finishing, are significant. There are also indirect costs associated with attrition. High turnover inevitably reduces morale and teamwork, both of which impact the long-term productivity of a group. Whitehead has been insulated from attrition by a bad economy, but it may lose personnel to industry when the economy recovers. The second reason to consider financial incentives is far more mercenary. When senior finishers are two to three times as productive as junior finishers, incentivizing them through overtime or salary increases may be sufficient to boost their capacity beyond that enabled by another junior finisher.

In short, Whitehead's incentive options are limited but critical to encouraging the kinds of performance it requires from its finishers. Finishers will not respond to the same incentives that inspire professional researchers. Whitehead's management must consider more creative uses of non-financial incentives. At the same time, it must also recognize the motivational and competitive importance of financial incentives. The costs for not doing so are significant.

8.5 Summary

In complex, analytical professions like finishing, much of the performance discrepancy observable between analysts can be attributed to differences in their natural aptitude. At Whitehead, organizational dynamics have unfortunately served to amplify those differences. Individual contribution continues to be the formal and informal focus of the organization, while teamwork remains a sought-after but institutionally unsupported objective. Teamwork, however, is critical to driving out process variability and boosting long-term productivity. For teamwork to become a reality at Whitehead, finishers must see a clear link between their daily behaviors and year-end rewards. That link can be made explicit through a performance review process that recognizes teamwork as a core value; a periodic, disciplined review of employee objectives; and an incentive system that recognizes the unique motivations of production personnel.

9 Conclusion

In many ways, data analysis production lines represent the vanguard of a new, service-oriented economy. Unlike traditional production lines, whose objective is a physical product, data analysis processes produce value-added information. Traditional concerns about the staging and consumption of physical materials are absent. They are replaced by complex information-processing tasks that must typically be handled by skilled human analysts. These differences have a profound impact on how data analysis processes are monitored and controlled. Variations in task complexity and analyst skill lead to higher process variability than that typically seen in traditional production processes. Managers, in turn, must focus more of their energy on gaining control of these processes.

The Finishing process at Whitehead is an excellent case study in the unique problems faced by managers of data analysis processes. Finishing tasks are inherently complex, leading to an iterative, trial-and-error process that is highly variable. This variability is heightened by large, persistent performance discrepancies that exist between finishers. Inefficient workflow policies have further aggravated the problem by lending finishers wide discretion in their activities. These three factors – high task complexity, varied skill levels, and inefficient workflow policy – have made it exceedingly difficult to predict and plan productive capacity. For the Human Genome Project, these process control problems had significant consequences: the group met its objectives but only after a sustained push that strained relations between finishers and management.

This thesis has sought to demonstrate that, although data analysis workflows are inherently variable processes, managers nonetheless have a number of valuable levers with which to temper their process control problems. Chapters 4, 5, and 6 identified workflow policies that reduce process variability stemming from how tasks are assigned to and managed by analysts. Chapters 5 and 7 analyzed team structures and the need to balance optimal task allocation against training and knowledge sharing opportunities. Finally, Chapter 8 analyzed the impact that organizational dynamics can have on analysts' motivation to work together and function as a team.

In summary, the key findings of this thesis are:

- **Break complex projects into simpler tasks.** Grouping multiple tasks into a single project boosts cycle times and increases output variability. To the extent possible, individual tasks should be assigned, tracked, and completed independently. Cycle times and variability will be reduced, making it easier to predict and plan capacity. The group will also be better positioned to study tasks and identify best practices, a capability that is masked when tasks are not tracked independently.

- **Assign tasks only when analysts have nothing else to do.** A just-in-time task assignment policy ensures high analyst utilization, while reducing WIP levels, cycle times, and process variability. Analysts that are allowed to build up large queues of work tend to prioritize that work according to short-term objectives. Judicious, timely task assignment eliminates their opportunity to do so.

- **Require analysts to process tasks in FIFO order.** When analysts have the discretion to prioritize their work, they tend to favor simpler tasks. FIFO processing ensures that tasks are processed in a timely fashion. Project cycle times are reduced and productivity becomes more predictable. FIFO processing also prevents analysts from building up large queues of difficult tasks.

- **Assign tasks to analysts based on their relative performance advantage.** Advanced analysts should be assigned difficult tasks only when they possess a relative performance advantage in those tasks. Managers, however, should actively monitor this relative advantage. If it proves illusory, assigning complex tasks to advanced analysts may actually harm the group's overall productivity. Managers should also balance optimal task assignment against the educational opportunities enabled by occasionally assigning difficult tasks to junior analysts.

- **Use team structure and task exchange to drive collaboration.** Data analysis is an inherently individualistic enterprise. When analysts labor in isolation, however, important opportunities for training and knowledge sharing are lost. A team structure that forces analysts to exchange tasks ensures that best practices are broadly disseminated and that skills are transferred from senior to junior personnel.

- **Make teamwork a central focus of the incentive system.** Structural changes to the organization are necessary but not sufficient conditions for getting analysts to think and act as a team. Teamwork must be rewarded by a company's performance review and incentive system. Only when analysts see a clear link between team-oriented behavior and personal rewards will they invest time into collaborating with their peers.

Data analysis production lines present unique process control problems. The study of one such production line, Whitehead's Genome Finishing Group, has yielded valuable insights into how these problems may best be managed. While the policies detailed in this thesis are far from complete, they establish a new baseline for controlling data analysis processes. These policies eliminate unnecessary sources of variability, freeing managers to focus on the production issues that matter most: scheduling, planning, and leading.

With the completion of the Human Genome Project in April 2003, Whitehead's Finishing Group at last finds itself in a position to consider long-term, strategic changes to the organization. Several of the changes proposed by this thesis have been implemented or are under serious consideration. Finishing projects are now being assigned in an objective manner that eliminates the conflict of interest that existed when coordinators controlled the process. Difficult tasks are being preferentially assigned to experienced finishers. And, finishers must now seek advice on projects that are not completed within a fixed number of work cycles.

With work just beginning on the mouse genome, Whitehead is also implementing the process and technology changes necessary to isolate and assign individual gaps from within BACs. This change is considerably more difficult than some of the others. Whitehead's entire process – indeed, even the exchange of information between genome centers – is geared towards the processing of BACs, not gaps. Nonetheless, Whitehead's management considers a gap-centric model of finishing to be an important, strategic change that will improve their ability to monitor and control the finishing process.

While these changes are underway, many of Whitehead's biggest challenges lie ahead. Analyst discretion will continue to be a major source of process variability until workflow policies such as just-in-time task assignment and FIFO work ordering are implemented. These policies represent a significant cultural change for an organization that has historically lent its analysts wide discretion. Likewise, Whitehead must overcome many organizational hurdles as it attempts to reduce the large performance discrepancies that exist between analysts. New team structures and better training programs are part of the solution, but these will pose significant cultural and incentive challenges for an organization that has traditionally focused on individual contribution.

Whitehead's efforts are significant not just because they impact the burgeoning and important field of genomics, but also because they are emblematic of the issues facing many businesses today. Success in an increasingly service-oriented economy requires managers that can lead despite the high process variability associated with information-processing tasks. The finishing process at Whitehead offers valuable insights into this endeavor.

10 Bibliography

Ancona, Kochan, Scully, Van Maanen, and Westney, *Organizational Behavior & Processes: Managing for the Future*, 1999, South-Western College Publishing.

DNA from the Beginning: an Animated Primer on the Basics of DNA, Genes, and Heredity, Cold Spring Harbor Laboratory, http://www.dnaftb.org/dnaftb/, 2002.

Drucker, Eccles, Ness, Cucuzza, Simons, Dbvlla, Kaplan, Norton, and Davila, *Harvard Business Review on Measuring Corporate Performance (Harvard Business Review Series)*, September 1998, Harvard Business School Press.

Gans, N., Koole, G., and Mandelbaum, A., "Telephone Call Centers: A Tutorial and Literature Review," OBP research group of the Vrije Universiteit Amsterdam, http://www.cs.vu.nl/obp/callcenters/, March 17, 2003, also appearing in *Manufacturing & Service Operations Management.*

Koole, G. and Mandelbaum, A., "Queueing Models of Call Centers: An Introduction," *Annals of Operations Research*, 113:41–59, 2002.

The Science Behind the Human Genome Project: Basic Genetics, Genome Draft Sequence, and Post-Genome Science, Office of Science, U.S. Department of Energy, http://www.ornl.gov/TechResources/Human_Genome/project/info.html, March 14, 2003.

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

11 Author's Biography

The author graduated from Princeton University in 1995 with a B.S.E. in Computer Science and a certificate in East Asian Studies and Japanese. From 1995 to 1996, he studied computer science and Japanese at the University of Aizu in Japan under a Fulbright Fellowship. Returning to the U.S., he researched digital television, computer graphics, and display technologies at Intel Corporation in Santa Clara, California for three years. From 2000 to 2001, he managed a team of software engineers at an interactive television company called ReplayTV. Finally, in 2001, he returned for his MBA and MS-EECS at MIT/Sloan, where this thesis was completed. The author can be reached at his permanent email address: scottr@sloan.mit.edu.