

**DESIGN OF AN AUTOMATIC DEFECT TRACKING AND YIELD
REPORTING SYSTEM FOR THE MANUFACTURING FLOOR**

by

Edward Matthew Lee

B.S., Mechanical Engineering, GMI Engineering & Management Institute

Submitted to the Department of Mechanical Engineering
and to the Sloan School of Management
in Partial Fulfillment of the Requirements for the Degrees of

Master of Science in Mechanical Engineering

and

Master of Science in Management

at the
Massachusetts Institute of Technology
June 1994

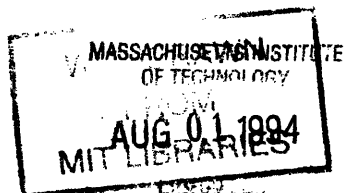
© Massachusetts Institute of Technology, 1994

Signature of Author _____
Department of Mechanical Engineering
Sloan School of Management
May 6, 1994

Certified by _____
Kevin Otto
Assistant Professor of Mechanical Engineering
Thesis Supervisor

Certified by _____
Roy E. Welsch
Leaders for Manufacturing Professor of Statistics and Management Science
Thesis Supervisor

Accepted by _____
Ain A. Sonin
Graduate Officer, Mechanical Engineering Department



DESIGN OF AN AUTOMATIC DEFECT TRACKING AND YIELD REPORTING SYSTEM FOR THE MANUFACTURING FLOOR

by

Edward Matthew Lee

Submitted to the Department of Mechanical Engineering
and to the Sloan School of Management
on May 6, 1994
in Partial Fulfillment of the Requirements for the Degrees of
Master of Science in Mechanical Engineering
and
Master of Science in Management

ABSTRACT

A project to design an automatic method for tracking and reporting yields and defects was undertaken at a division of the Hewlett-Packard Company. Before the project began, the company had been manually collecting performance data, and automatically collecting cycle-time information on the production floor for a discreet component manufacturing area. Yield reports were manually constructed from this data. The process of constructing these reports was somewhat tedious, and accuracy issues had surfaced. This situation, combined with a rapidly changing business environment, necessitated the project. All of the materials were provided by the Hewlett-Packard Company.

The new system uses bar-code readers to enter production data into a centrally-located, relational database. A microcomputer-based application program was constructed to extract information from the central database, perform calculations on the data, and report the results to the user in both on-screen and hardcopy formats. The nature of the database searches are defined by the user, and the results are provided real-time. Various performance metrics, including First-Pass Yield and N-Pass Yield, can be returned by the system. New performance metrics were also defined that take advantage of the new reporting system. These new performance metrics, Root Cause Yield and Average Number of Passes, can also be calculated automatically. In addition, Pareto Charts of defects can be created.

The thesis report is divided into three chapters. In the first chapter, the problem is introduced. In the second chapter, conclusions are given, as are recommendations for further study. In the third chapter, a detailed explanation of the methods used for the investigation are presented.

Supervisors:

Kevin Otto, Assistant Professor of Mechanical Engineering
Roy E. Welsch, Leaders for Manufacturing Professor of Statistics and Management Science

PREFACE

This thesis was written to fulfill the thesis project requirement for the Master of Science in Mechanical Engineering and the Master of Science in Management Degrees at the Massachusetts Institute of Technology. It is the culmination of two years of work in the Leaders for Manufacturing Program. The program combines academic instruction in the classroom with practical experience in the workplace. The two years were divided between semesters at school and an internship at a company site, providing for a year-round educational process. The investigation and analysis of the problem discussed in this paper took place during a six and one-half month internship at Hewlett-Packard. It should be noted that a project of this scope and scale would have been very difficult, if not impossible, without the combination of managerial and engineering instruction provided by MIT. The author wishes to acknowledge the Leaders for Manufacturing Program and the National Science Foundation for their support of this work. Particular appreciation should also be given to the Medical Products Group of the Hewlett-Packard Company for the opportunity to carry out this investigation.

Several people were instrumental to the completion of this thesis by providing valuable information, assistance, and support. Their help proved to be essential to me throughout various phases of the project. In particular, the following individuals provided significant contributions:

1. Martha Keller, Manager, Component Assembly Department, Hewlett-Packard Company, helped me every step of the way and served as the project advisor at the company site. She helped with the formulation of the investigation, provided guidance throughout the study, supplied critical feedback, and aided in the implementation.
2. Dexter Daniel, Engineer, Wire and Test Department, Hewlett-Packard Company, provided enormous support and information during every phase of the investigation. He laid the groundwork for the investigation and contributed fully half of the program code for the resulting system. Without Dexter, this thesis project could never have been realized.
3. Kate Stohlman, Manager, Component Manufacturing Section, Hewlett-Packard Company, initiated the project and provided necessary direction for the work. Kate acted as guide, advisor, and mentor.
4. Stu Siegel, Manager, Component Manufacturing Engineering, Hewlett-Packard Company, provided additional guidance, feedback and support. Stu's vision for the system kept the investigation focussed and moving toward the final goal.
5. Kevin Otto, Assistant Professor of Mechanical Engineering, Massachusetts Institute of Technology, served as my thesis advisor from the engineering school. Kevin provided suggestions that led to new findings and a more well thought out analysis.
6. Roy Welsch, Professor of Management Science, Massachusetts Institute of Technology, served as my thesis advisor from the Sloan School of Management. Roy provided solid feedback on the program's functionality, as well as guidance and suggestions for addressing the organizational issues presented in the project.
7. Cecil Stackpole, Associate Professor of Communication, GMI Engineering and Management Institute, provided guidance for the organization and format of this thesis.

ADDITIONAL COPYRIGHT INFORMATION

References to various copyrighted software packages are made in this document. The appropriate copyright references are made below.

- Microsoft® is a registered trademark of Microsoft Corporation
- Windows™ is a trademark of Microsoft Corporation
- Microsoft Access™ is a trademark of Microsoft Corporation
- LANManager™ is a trademark of Microsoft Corporation
- Borland® is a registered trademark of Borland International, Inc.
- Paradox® is a registered trademark of Borland International, Inc.
- Approach® is a registered trademark Approach Software Corporation, a wholly-owned subsidiary of Lotus Development Corporation

TABLE OF CONTENTS

ABSTRACT.	2
PREFACE.	3
ADDITIONAL COPYRIGHT INFORMATION.	5
LIST OF ILLUSTRATIONS	9
I. INTRODUCTION	11
BACKGROUND	13
The Manufacturing Process	14
The Component Manufacturing Area.	15
STATEMENT OF THE PROBLEM	17
Limitations of the Logbooks	17
Limitations of the QIC System	18
Limitations of the Reporting System	20
Summary of the Problem	27
OBJECTIVES OF THE INVESTIGATION	27
SIGNIFICANCE OF THE PROBLEM.	30
Significance to Hewlett-Packard Company	31
Significance to the Leaders for Manufacturing Program and MIT	32
Significance to the Thesis Author.	33
SCOPE AND LIMITATIONS OF THE PROJECT	33
OTHER BACKGROUND INFORMATION.	34
METHODS AND MATERIALS FOR INVESTIGATION.	35
PREVIEW OF THE REST OF THE REPORT.	36

II. CONCLUSIONS AND RECOMMENDATIONS	38
PERFORMANCE METRICS.	38
First-Pass Yield	39
What is It?	
Why Measure It?	
N-Pass Yield	42
What is it?	
Why Measure It?	
Terminate Yield.	44
What is It?	
Why Measure It?	
Root Cause Yield	47
What is It?	
Why Measure It?	
Average Number of Passes	49
What is It?	
Why Measure It?	
Summary of Process Metrics	52
THE YIELD AND DEFECT REPORTING SYSTEM.	52
The Nature of the Data Collection Scheme	53
Collection and Reporting Schemes	
Outside vs. In-House Development	
Database Infrastructure	61
Database Architecture.	62
The UNITS Table	
The ACTIVITY Table	
The DEFECTS Table	
The HISTORY Table	
Use of the System - Entering Data	
Other Tables	
The Yield Reporting and Defect Tracking Application.	70
RECOMMENDATIONS FOR FURTHER ANALYSIS.	73
System Ownership	73
Database Size	74

III. DISCUSSION 75

BACKGROUND OF EXISTING DATA INFRASTRUCTURE. 76

INFORMATION GATHERING 78

 Informal Interviews 79

 Surveys 80

 Operator Yield Reporting Survey

 Engineer Yield Reporting Survey

 Time Survey

 Programming Tools 84

PROTOTYPE BUILDING 86

 Refining Customer Needs and Expectations 87

 Pareto Chart building

 Definitions of Yield Numbers

 Root Cause Defect Map

 Usability Feedback 92

 Programming Skill Practice. 93

PROTOTYPE REFINEMENT AND SOLUTION GENERATION 94

 Solution Generation 94

 Implementation 96

 Final Project Analysis. 97

BIBLIOGRAPHY. 100

APPENDICES 101

LIST OF ILLUSTRATIONS

FIGURES

1.	Process Flow in the Manufacturing Area	14
2.	Illustration of the Creation of Downstream Defects	19
3.	The Process of Generating the Manual Yield Report	21
4.	Sample Section of the Manual Report	22
5.	Calculations for Terminate and N-Pass Yields	26
6.	The Division's Information Strategy for the Manufacturing Floor	31
7.	Description of the Flow of this Document	37
8.	Manufacturing Hierarchy	55
9.	Manufacturing Information Strategy Matrix	56
10.	Diagram of the Infrastructure of the Information System	61
11.	The Four Main Tables in the Yield and Defect Reporting System	63
12.	Progress of a Unit Through the Manufacturing Process	68
13.	Second Half of the Sample Unit's Progress Through the Manufacturing Area	69
14.	Program Manager Screen	71
15.	The Opening Screen of the MARS Program	72
16.	The Main Menu for the Mars Program	72
17.	Sample Flow of a Defective Unit	88

18. Pareto Screen for the MARS System 90

TABLES

I. Correct and Incorrect Method for Determining Composite Yield 24

II. Comparison of Yield Results from Different Reports 28

III. Explanation of Information Strategy Matrix 57

IV. Sample Record from the History Table 64

V. Outline for Investigation 76

VI. Questions Regarding Manufacturing Data Collection 80

VII. Top Eight Comments Returned On Technician
Yield Reporting Survey 81

VIII. Comparison of Old System to MARS Project System 97

I. INTRODUCTION

In the manufacturing environment, speed has become a new measure of importance. In order to compete in a global economy, production cycle times have to be reduced, time to market has to be improved, and customer response times must be shortened. If companies don't follow this advice, their speedier competitors will rush past them to greater financial rewards. Much attention has been given to using information technologies to help companies achieve this new level of speed.

However, heavy investments in information technology have delivered disappointing results. In turn, these results have prompted some companies to stop any further investments in information technology. According to managers at these companies, the investment just hasn't been worth it. Michael Hammer, president of Hammer and Co., an independent information technology consulting firm in Cambridge, MA, believes that this disappointment stems largely from the way the new technology is used. In his view, companies tend to use technology to mechanize the old ways of doing business. They leave the existing processes intact and use computers simply to speed them up.¹

But speeding old processes cannot address their inherent fundamental deficiencies. Unfortunately, many of our current job flows, work designs,

¹Hammer, Michael, "Re-engineering Work: Don't Automate, Obliterate", *Harvard Business Review*, July-August, 1990, Page 104.

organizational structures, and control mechanisms came of age in a different competitive environment – before the advent of the computer. These processes are geared toward efficiency and control. Yet the buzzwords of today are innovation and speed, service and quality. Instead of "paving cow paths," by overlaying new technology on old processes, Hammer suggests that we obliterate our outdated processes and start over. To use a term that he coined, we should "re-engineer" our business. That is: use the power of modern information technology to radically redesign our business processes in order to receive dramatic improvements in their performance.

A division of Hewlett-Packard Company recognized, last year, that they needed to change the way in which production performance information was collected and distributed to the members of the manufacturing areas. While a solid performer in their field, they faced formidable competition from focussed domestic players, as well as large foreign competitors. The market for medical technology has been changing rapidly, posing new challenges to all areas within the company. HP understood that faster information flow in the manufacturing area could help them to respond to their changing external environment.

This thesis, "Design of an Automatic Defect Tracking and Yield Reporting System for the Manufacturing Floor," was brought about by the need to re-engineer HP's manufacturing information system. My work at the company site involved understanding the problems that they were having with the current system, determining the needs of all of the parties involved, and designing a new system to

track and report production performance information. Following Hammer's advice, I chose to institute a new system instead of merely automating the old one. I hope that this paper can be used as a guide for those attempting to employ computers to track manufacturing data, or by others who are trying to re-engineer different processes within their companies.

BACKGROUND

Organizationally, the division is part of the Medical Products Group within Hewlett-Packard. The Medical Products Group builds a wide variety of high-tech medical equipment, primarily for use in hospitals and larger clinics. Their products range from bedside monitors for intensive care units, to hospital-wide information systems, to stethoscopes. The products, like those built by other high-tech companies, are low volume, high mix, high precision devices, constructed of high value-added materials. To accomplish the difficult task of building these products, HP employs a workforce of highly skilled technicians. In addition, HP has a goal to implement self-directed work teams on the manufacturing floor. Accurate, up-to-date information has to be available to everyone on the floor, in rapid fashion, for groups of technicians to be able to function as self-directed teams.

The Manufacturing Process

The manufacturing process at the division is divided into five main parts, as shown in Figure 1 below. The Component Manufacturing Area builds sub-

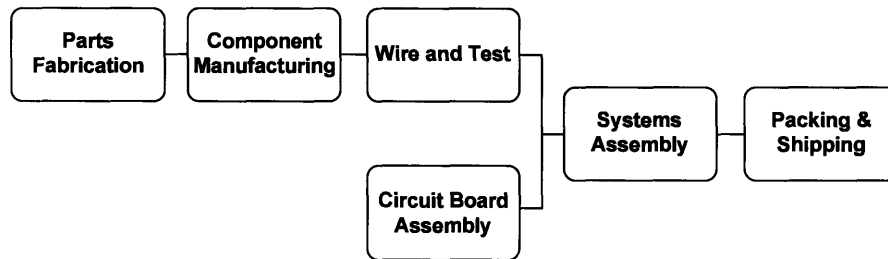


Figure 1. Process flow in the manufacturing area.

assemblies that are later wired into a completed assemblies. This wiring takes place in the Wire and Test Area. While these products are being built, circuit boards are constructed. At the System Assembly Area, completed electronic components and circuit boards are assembled into finished systems. The systems are then packaged and shipped to customers. The bulk of this paper focusses on the unique problems of the Component Manufacturing Area, although the results and conclusions are applicable to the entire manufacturing process. Also, because of the proprietary and sensitive nature of the actual manufacturing process at Hewlett-Packard, some of the examples in this paper will use the construction of a toy car to illustrate the various assembly steps in the process.

The Component Manufacturing Area

Before the investigation began, the Component Manufacturing Area tracked three primary types of data on the manufacturing floor. These were the number of good units produced per day, the defects associated with rejected units, and the cycle times associated with terminated units.² The number of good units was used to formulate yield figures. These yield figures, in turn, were used to evaluate the performance of the manufacturing line, while the defect information was used to direct improvements in the process. HP has a well-established continuous improvement process.

The company had a variety of methods in place to handle the problem of yield and defect tracking and reporting. At every process step, the technicians logged each unit into a logbook as it entered the operation. As the units left the operation, they were logged-out either as *good parts* or *scrap*. By dividing the number of good units by the number of total units started, each technician could calculate a "first-pass yield" for the day. This yield number was useful in that it gave the technicians an idea of "how the process went" that day. These logbooks were located at every operation step.

In addition to the logbooks, where a lot of manual data was collected, there was an automatic tracking system called QIC. The QIC (**Q**uality **I**mprovement using

²For the purposes of this investigation, I will use the expression *terminated* to indicate units that have exited the manufacturing process, whether as acceptable units or as rejected units. The term *scrap* will often be used to denote rejected units.

Cycle Times) System was put into place four years prior to help the company report the cycle times of their parts. The system used bar-code readers to scan each unit as it entered and exited the manufacturing area. A line of information was entered into a database when each unit was scanned-in. The units were identified in the database by a unique Unit Number. Each piece of data was then updated with the cycle time when the particular unit was scanned-out. The QIC System was driven by a dedicated PC that calculated the cycle times and managed the transactions. At the end of the month, all of the entries in the database table were averaged by part number, the results of which were printed in a paper report. If a unit was scrapped, the QIC System stored a "defect code" and a brief description of the defect, along with the unit number, in a separate table in the database. A more detailed description of the QIC System is given on page 76.

In addition to the logbooks and QIC, a stack of information sheets called the "fab file" was routed with each unit as it proceeded through the assembly process. The technicians in the manufacturing area had to sign their initials to the fab file at each step, indicating that their particular operation had been successfully completed for each unit. The fab file contained information on the component lot numbers added at each step, test results, and other engineering data. The Food and Drug Administration required much of the information contained in the fab files.

Once a month, several individuals gathered data from the logbooks and the QIC System to form a yield summary for the manufacturing floor. Then, yield reports were generated by breaking down the yields for each part number and operation and

typing the results into a spreadsheet. Pareto Charts of defects were constructed partially by manual counting, and partially through the execution of spreadsheet macros.

STATEMENT OF THE PROBLEM

The manual data collection and semi-automatic report generation served HP for a number of years. The system allowed for the smooth introduction of new products, the successful completion of process improvements, and the reduction of cycle times. Engineers and managers could count on regular reports of process performance. Dedicated line support personnel maintained and improved the system over time. However, there were still a few limitations to the system. Below are some of the more major limitations that sometimes made it difficult to get accurate, timely information from the system.

Limitations of the Logbooks

The logbooks mentioned in the last section were located at each workcenter at various locations spread-out over the entire manufacturing floor. This meant that the line support personnel involved in calculating the yields at the end of each month had to spend a large amount of time on the floor collecting data from the different logbooks. Sometimes, the data was incomplete, requiring the coordinators to spend

even more time asking various technicians to "fill in the gaps" in the data. Indeed, due to the increasing complexity of the products, gaps in the logbooks were becoming more and more frequent. The line support personnel sometimes complained that their job was getting tougher as time went on.

Limitations of the QIC System

Units were scanned-out of the QIC System for one of two reasons – either as good units or scrapped units. When a unit was scanned-out as a scrapped unit, the technician also stored a defect code. The four-digit defect codes incorporated a two-digit defect identifier and a two digit operation number to help identify the location in the process where the defect occurred. Then, at the end of the month, the person compiling the data would be able to tell which operation had the highest incidence of scrapped units. But, while storing the operation number where the unit was scrapped was appropriate in most cases, there were many instances where the defect was *discovered* at an operation that was different than the operation where the defect was *produced*. As a result, an operation was often "charged" for a defect which did not originate at that operation. For example, a defect caused by a faulty part attached at Operation 10 that is later discovered at Operation 20 will be counted as an "Operation 20 defect," instead of an "Operation 10 defect." As a result, Operation 20's yield will be artificially lowered, even though the defect was really attributable to Operation 10's process. Figure 2 on the next page provides an illustration of this point. The

front bumper is attached to the car at Operation 10. Later, while the technician at Operation 20 is attempting to attach the rear bumper, the front bumper falls off, causing the car to be scrapped. The yield for Operation 20 will be lowered, even though the cause of the problem was that the technician at Operation 10, let's assume,

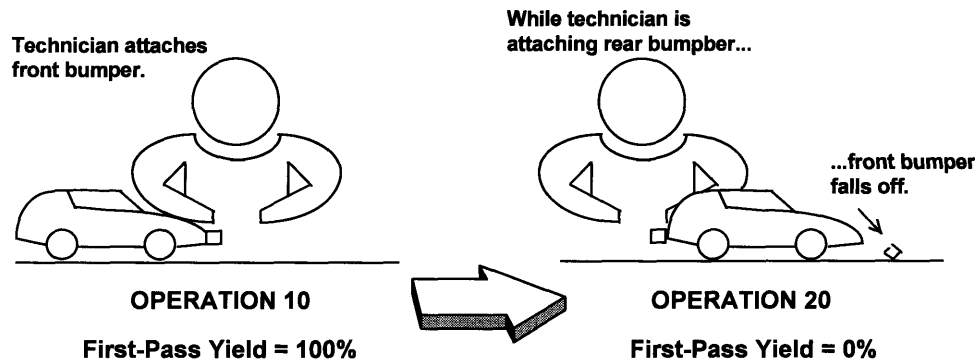


Figure 2. Illustration of the creation of downstream defects.

used an incorrect amount of torque on the screws. Assume, also, that this was a relatively common occurrence at a few of the operations. Not only did this situation lead to inaccurate reporting of yields for Operations 10 and 20, but there was no formal mechanism in place to feed the information about downstream defects back to the originating operation.

Another limitation of the QIC System stemmed from the fact that only two "snapshots" were taken of each unit that passed through the manufacturing area – one at the beginning, and one at the end. As such, QIC didn't allow for the tracking of a unit as it progressed through each operation in the area. To get information for each operation, a person had to manually get the data from the technician's logbooks.

Furthermore, the defect codes that were entered into the system were seldom

more than failure modes, not detailed descriptions of why a unit failed. While many units were torn-down by technicians to determine the root cause of the defect, the QIC System still only stored the original failure code; even if the root cause analysis unearthed a different reason for the failure. Therefore, any Pareto Charts generated from the QIC System were somewhat inaccurate, in that they didn't contain the most up-to-date data.

Limitations of the Reporting System

Two yield reports were produced each month. One was produced by the line support personnel, who collected yield data from the logbooks. This was known as the *manual report*. Another report, the *semi-automatic report*, was produced by an engineer who had ownership for the QIC System. Her job was to execute some spreadsheet macros that converted the data from Paradox database format into Lotus 1-2-3 spreadsheet format. Each report was distributed to process engineers, supervisors, and managers, and each report had its own unique limitations.

The manual report was extremely difficult to generate. It required the line support personnel to follow the process outlined in Figure 3 on page 21. The process was very tedious, since the technician's logbooks were scattered throughout the manufacturing floor. Completion of the report took two coordinators approximately two full days to gather the information, check it for completeness and accuracy, and generate the document. A sample of the document is given in

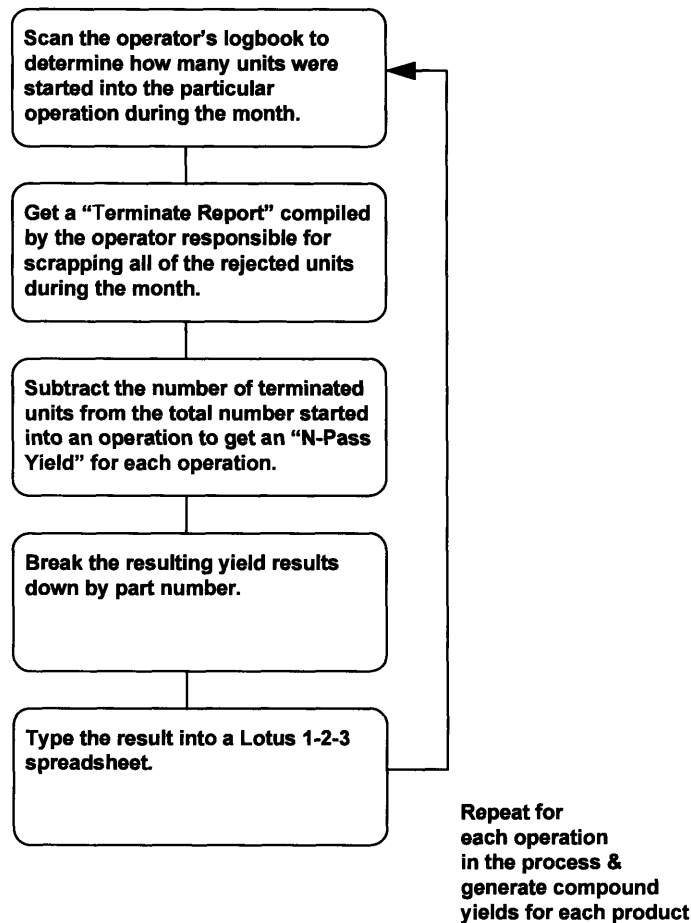


Figure 3. The process of generating the manual yield report.

Appendix A on page 102.

Note that the third box from the top of Figure 3 mentions the term "N-Pass Yield." This term was generally understood to indicate the total number of good completed units built during a specific time period, divided by the total number of starts for a time period. This differs from "First Pass Yield" in a number of ways. See the section titled "Performance Metrics" on page 38 for more precise definitions of the yield numbers.

TOY CAR MANUFACTURING MAY 1993												
	Car Model A			Car Model B			Car Model C			Car Model D		
	Good	Bad	%	Good	Bad	%	Good	Bad	%	Good	Bad	%
Attach Frt. Bumper	4	1	80	10	1	91	11	1	92	12	2	86
Attach Rear Bumper	5	2	71	15	1	94	12	2	86	14	3	82
Attach Doors	6	1	86	18	2	90	13	1	93	16	2	89
Attach Hood	7	2	78	13	2	87	15	1	94	13	1	93
Composite Yield	38%			66%			68%			58%		
Summary												
Total throughput	48											
Weighted Average Yield	61%											

Figure 4. Sample section of the manual report.

Figure 4 above gives a sample section of the report. The section is for four different part numbers of cars. The number of units that were determined to be "good" or "bad" for each operation and part number is given. Note that the yield percent was determined by dividing the number of good units by the number of total units that passed through the particular operation in the process. This is consistent with the definition of N-Pass Yield above.

The Composite Yields in Figure 4 were determined by multiplying the different N-Pass Yield percentages together for each operation. Then, a Weighted Average Yield was determined by multiplying the Composite Yield for each part number by its share of the total throughput for the month. Both the Composite Yield number and the Weighted Average Yield number are misleading, however.

Because it takes considerably less than a month for a unit to pass through the entire process, and because the yield report is simply a monthly snapshot of the

process, a lag is introduced between each operation. This may mean, therefore, that the four units that are accounted for in the first operation for Model A in Figure 4 are not necessarily the same four units that are represented in each subsequent operation. For example, while seven units went through "attach doors" on May 15, seven completely *different* units passed through "attach hood." Because of this, the term "Composite Yield," as used in this context, is inaccurate. One can only use the term "Composite Yield" when the same group of units is referred to at each operation. An example of this is given in Table I on page 24. Note that the ten units in the first table are followed through the *entire* process. In the final analysis, there were ten units started into the manufacturing area in May, of which two were successfully shipped to customers. The total Composite Yield was, therefore, 20 percent. However, in the second table, two separate groups of ten units are followed through the process. By this table, ten new units entered the process in May, and two are likewise shipped to customers, but the Composite Yield is calculated as only 14%. This is due to the fact that there were other units (most likely started into the process in April), that confounded the calculation. This is an illustration of a significant limitation in the manual yield report. Different units, that are started in different months, are included in the Composite Yield calculation for a single month, giving incorrect results.

The phenomenon of the Composite Yield inaccuracy was further disturbing in light of the fact that "Total Process Yield" was included at the bottom of the *manual report* that was the product of all of the Composite Yields for each part family. (Refer, again, to Appendix A on page 102.) This Total Process Yield was the

**Table I
Correct and Incorrect Method for
Determining Composite Yield**

	Date	Operation	Total Starts	Number Good	Number Bad	% Yield
Correct Method	5/1	Attach Body	10	9	1	90%
	5/10	Attach Wheels	9	7	2	78%
	5/20	Attach Doors	7	4	3	57%
	5/30	Attach Bumpers	4	2	2	50%
	TOTAL		10	2	8	20%

	Date	Operation	Total Starts	Number Good	Number Bad	% Yield
Incorrect Method	5/1	Attach Body	10	9	1	90%
	5/10	Attach Wheels	9	7	2	78%
	5/1	Attach Doors	10	4	6	40%
	5/10	Attach Bumpers	4	2	2	50%
	TOTAL					14%

number that was generally understood to represent the overall performance of the manufacturing area, and it was closely followed by upper management. However, as one can see, the Total Process Yield was created by multiplying several inaccurate numbers, namely the various Composite Yields, together.

Furthermore, the monthly snapshot provided by this report was inadequate because it had no way of giving trend information over the course of a month. A monthly yield of 50% for an operation may mean that the process had a 50% yield every day for 30 days, or it may mean that the operation had a yield of 0% every other

day, and 100% every other day. The engineer responsible for each operation generally knew this day-by-day information, but it wasn't documented in the yield reports. Lastly, the time consumed in generating the *manual report* made the possibility of obtaining more than one report per month fairly unrealistic.

Earlier, I mentioned the other *semi-automatic report* that was generated from the QIC data. Like the *manual report* dissected in the preceding few paragraphs, the *semi-automatic report* contained some inadequacies. A copy of this report is given in Appendix B on page 103.

Like the *manual report*, it was also tedious to generate, and therefore only produced on a monthly basis. As such, it didn't give insight into daily performance.

The QIC System took information in the form of a "defect code" from technicians whenever a unit had to be scrapped. A Pareto Chart could then be generated from data on the QIC System. However, the format for this Pareto Chart was difficult to read. Note the bottom half of the report in Appendix B on page 103. The Pareto Chart is certainly there, but in a rather difficult and non-conventional format.

Perhaps the most glaring shortcoming of this report was that it didn't give any information for the specific operations in the manufacturing area. This was because the QIC System only obtained two pieces of information about each unit – the starting point and the termination point. Data was not entered into the QIC System for each operation in-between.

It may be worth noting that the yield numbers represented in the *semi-*

automatic report were known as "Terminate Yields." That is, the yields were calculated from the units that were terminated³ from the manufacturing process for the time period of the report. The resulting numbers were very different from the "N-Pass Yields" depicted in the *manual report*. Figure 5 shows the difference in calculations for the two numbers. Needless to say, the yield numbers contained in the

$$\textit{Terminate Yield} = \frac{\textit{number of good units}}{\textit{number of total terminates}}$$

$$\textit{N-Pass yield} = \frac{\textit{number of good units}}{\textit{number of starts}}$$

Figure 5. Calculations for Terminate and N-Pass Yields. Note the difference.

manual report did not match those contained in the *semi-automatic report*. Table II on page 28 shows the resulting yield numbers for the month of May, 1993 from the two reports. The numbers represented in the two reports were actually the result of two fundamentally *different* calculations. The fact that the numbers from the two reports did not agree served to lessen the credibility of both reports. Many engineers calculated their own yield numbers when they needed to have yield information, instead of relying on the reports. When surveyed about how much time engineers spent calculating yield numbers, the average number returned was three hours per week per support engineer.

³Recall that the term *terminated* in this context means "exited the system." A terminate may be a successfully shipped "good" unit, or an unsuccessful, scrapped "bad" unit. Within a given time frame, the number of terminates differs from the number of starts by the amount of work in process.

Neither the *manual report* nor the *semi-automatic report* contained information about rework events. When a unit was reworked for a specific defect, the defect description and the outcome of the rework event were only tracked in the fab file, not on QIC *or* in the logbooks. As a result, assembling a monthly summary report of rework information would require an individual to scan all of the fab files for every unit produced over the month. Since this would be a truly grueling task, it was rarely done.

Summary of Problem

I have detailed the major problems with the old method of reporting yields and tracking defects on the manufacturing floor at the division. This system, although flawed, served the company for many years. The reader is reminded that the system was entirely adequate until recently, when stronger demands from the market and increases in production volume and mix began to strain its foundation. However, it was clear to many that something needed to be done to change the way in which performance information was collected and reported.

OBJECTIVES OF THE INVESTIGATION

There needed to be a method by which managers, engineers, and technicians could get up-to-the minute information on the yields and defects for each operation,

TABLE II
Comparison of Yield Results from Different Reports
May, 1993

Part Number	Yield Reported in Manual Report	Yield Reported in Semi-Auto Report
Car Model A	38%	54%
Car Model B	66%	68%
Car Model C	68%	75%
Car Model D	58%	62%
Car Model E	38%	50%
Car Model F	61%	63%
Car Model G	38%	50%
Car Model H	10%	62%
Car Model I	70%	74%
Car Model J	55%	59%
Car Model K	57%	64%
Car Model L	92%	93%

**Numbers have been disguised.*

and for the manufacturing area as a whole. The primary purpose of my investigation was to re-engineer the process of obtaining and reporting performance information on the manufacturing floor. As such, I planned to design an information system.

The design of my system had to meet three goals. First, it had to be able to satisfy the needs of the customers; namely the technicians, engineers, and managers in the Component Manufacturing Area. Second, it needed to fit into the overall strategy for information systems at Hewlett-Packard. Third, the project needed to be

completed in the context of my six and one-half month internship.

There were three primary milestones in the progress of the investigation.

These milestones served as the objectives of my investigation, and are detailed here.

First, I had to determine the appropriateness of an automated data collection scheme. When data is being collected and manipulated in a manufacturing environment, there should be a good fit of the method to the conditions. In a low-volume environment, or in an environment where reporting is infrequent, manual collection schemes often prove to be adequate, and the expense of automatic, electronic systems is unwarranted. As volumes increase, or the need for more frequent data handling and reporting arises, the time involved in manually preparing reports may eventually outweigh the time needed to develop an automatic collection scheme. With high volume production, tracking each individual part may become unnecessary, and sampling becomes the most efficient way to collect meaningful data. I devoted some time to the investigation of the tradeoffs between the manual, automatic, and sampling data collection schemes, and the appropriateness of automatic data collection in this particular instance.

Second, as part of the investigation, I had to determine what information was genuinely useful and needed by the various personnel in the area. The process metrics employed had to pass a litmus test of usefulness. The definitions of these metrics also had to be solidified. Any additional metrics that allowed technicians to better monitor themselves, engineers to more easily prioritize work, and managers to allocate resources more effectively, needed to be identified.

Lastly, I endeavored to design a PC-based reporting program that would allow information to be retrieved from a central database quickly. When completed, a person seeking information from the database should be able to ask the system to query the data on a variety of parameters (part numbers, dates). The user should be able to interactively define these parameters, or criteria, for each search. The program should then be able to generate yield *snapshots*, as well as produce graphs of yield *trends*. It should also be able to generate Pareto Charts of defects. Furthermore, a friendly and efficient user-interface had to be designed to accomplish the task. A system that is difficult to use would not be used. I drew upon knowledge of user-interface design that I gained in coursework at MIT.

Taking these milestones into account, and the goals of re-engineering, the chief objective of my investigation was to develop a new system that shortened the time needed to get information on process performance, while improving the accuracy of the information.

SIGNIFICANCE OF THE PROBLEM

In general, by shortening the time needed to get information about process performance, while improving the accuracy of information presented, it becomes more readily apparent where process improvements can have the greatest impact. Successful resolution of the problems presented can have a positive effect on all the major parties involved in the investigation. The completion of this thesis, therefore,

has significance not only to Hewlett-Packard, but to MIT and others. The extent of this is detailed below.

Significance to Hewlett-Packard Company

In addition to solving the problems outlined in the previous few sections, there are other benefits to the company sponsoring the project. The division has a goal to completely automate and computerize data collection and retrieval on the manufacturing floor. The plan for this is outlined in Figure 6 below. The first step, the automatic calculation of cycle times, is already in place with the QIC System. The next step, computer tracking of yields and defects, will hopefully be accomplished by the project presented here. Then, the company can focus on getting all computer-

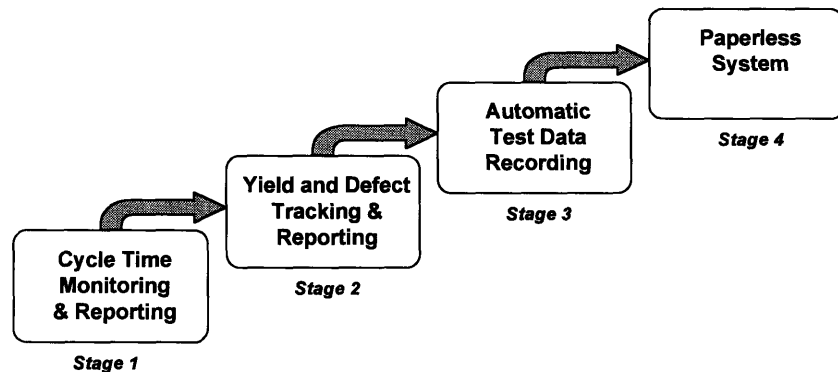


Figure 6. The division's information strategy for the manufacturing floor. This project embodies Stage 2.

generated test data out of the fab files and into a central electronic database. Lastly, the paper fab file system can be eliminated by having all data entered, stored,

manipulated, and retrieved via computer. This project, as one can see, is a vital step on the road to a paperless system at HP. With improved tools, such as the one proposed here, the company can streamline information flow in the manufacturing area, leading to better management and engineering decisions, and hopefully leading to a cost savings for the company.

Additionally, HP will gain a tool that will help the company to implement self-directed work teams by providing technicians with the ability to measure their own performance.

Significance to the Leaders for Manufacturing Program and MIT

With this project, LFM and MIT can strengthen their alliance with Hewlett-Packard. One way that this project accomplishes this goal is by improving the already strong relationship between the Medical Products Group and faculty at MIT. Two faculty members assisted in the definition and implementation of this project. Kevin Otto, Assistant Professor of Mechanical Engineering, provided guidance pertaining to the structure of the program and the construction of the user interface. Roy Welsch, Professor of Management Science at the Sloan School of Management, provided assistance with regard to data handling and manipulation, as well as guidance into other management issues related to the project. Both professors have guided the production of the actual thesis document.

The support of HP is vital to the survival and prosperity of the Leaders for

Manufacturing Program. A successful project that provides an economic return to the company can help to justify Hewlett-Packard's continuing participation in the LFM Program. With HP's help, LFM can potentially be a key force in the strengthening of manufacturing education in America.

Significance to the Thesis Author

This project provided me with the opportunity to gain experience tackling a complex problem in a manufacturing environment. I was given the rare opportunity to implement the ideas and knowledge gained from my classes at MIT in a controlled setting. I also gained new knowledge on database and network management, two fields that are becoming increasingly important in manufacturing.

SCOPE AND LIMITATIONS OF THE PROJECT

The PC application resulting from this project is concerned with yield and defect tracking and reporting. The actual data input tools were developed by other engineers working at HP. I have responsibly documented the contributions that I made to the best of my ability, so that future HP engineers can make the proper modifications to the work that I have performed as new needs arise.

I have limited the investigation primarily to the Component Assembly Department within the manufacturing area, building off of the QIC System hardware

that was already installed there. All units that pass through this portion of the manufacturing floor have their own "identities" in the form of bar-coded labels, thus simplifying the tracking effort. I did not specifically address the Parts Fabrication Department, nor did I concern myself with downstream processes. Instead, I attempted to attach appropriate handles to the program that should allow modules pertaining to these areas to be added later. The six and one-half month time period allotted for the investigation was the true limiting factor.

OTHER BACKGROUND INFORMATION

In the months before this investigation, I conducted informal interviews with members of the Component Manufacturing Department to help identify problems with the existing yield reporting methods. It was from these interviews that I was able to formulate the project.

The QIC System, in place at the division, stored data in Paradox-formatted database tables located on a network. Because the project was intended to build upon the database connected to the QIC System, I used the Paradox Application Language (PAL) to write the code for the application. (For more information on my selection of Paradox, please see the section titled "Programming Tools" on page 84.) I consulted two books of Paradox and PAL. The first was *Mastering Paradox for Windows* by Alan Simpson.⁴ This book helped me to understand the fundamentals of database

⁴Simpson, Alan, *Mastering Paradox for Windows*, (Alameda, CA: Sybex, 1993)

construction as well as the structure of Paradox tables. For help on the programming language, I consulted *Paradox for Windows Programming by Example*, by Diane Tinney.⁵ The author of the book does an excellent job of presenting ObjectPAL, the object-oriented version of the PAL language. Without this book, I would not have been able to begin this project.

Much of the investigation, especially the design of the user interface, will be conducted using methods taken from "Concept Engineering," a publication by Gary Burchill of the Center for Quality Management in Massachusetts.⁶ CQM consults companies on the product development cycle, and Concept Engineering is a technique for taking customer voices and translating them into designs.

METHODS AND MATERIALS FOR INVESTIGATION

In this section, I will give a skeletal overview of the methods that I used to conduct the investigation. A more detailed account can be found in Chapter Three, but I feel as though an overview will help the reader to better understand the conclusions presented in the next chapter. To begin, I conducted interviews and a formal survey at the site. Then, I used some of the tools of Concept Engineering to further identify and define the needs of the customer, and to formulate a preliminary

⁵Tinney, Diane, *Paradox for Windows Programming by Example*, (Carmel, IN: Que, 1993)

⁶"Concept Engineering", tools for listening to the voice of the customer, available from Gary Burchill, the Center for Quality Management, Massachusetts, 1992, or from the Leaders for Manufacturing Program at MIT, Cambridge, Massachusetts.

design. Concurrent with the construction of the preliminary design, efforts were made to develop and refine the metrics that were to be the output of the program. After reviewing the design with people at HP, refinements were made. The program was then implemented on a trial basis.

All of the materials for this project were furnished by the Hewlett-Packard Company. The investigation took place on-site at the HP division. HP gave me the use of a personal computer, along with all of the software and tools necessary to complete the project.

PREVIEW OF THE REST OF THE REPORT

It is hoped that this introduction has served to give the reader a solid understanding of the nature and significance of the problems faced by Hewlett-Packard. In it, I have tried to answer two questions. First, "What was the problem?" Second, "Why bother to solve it?" Armed with this answers to these questions, the reader should be ready with a third question, namely, "What is the solution?"

Immediately following this section is chapter outlining the conclusions that I reached, along with some recommendations for further study. The chapter is broken down into three main sections. The first section is designed to describe the new metrics that I developed to allow the key groups of customers (technicians, engineers, and managers) to gauge the performance of the manufacturing area and to pinpoint problems. The second section is designed to provide an overview of the tracking and

reporting system that I developed to deliver the new metrics to the key customer groups. The last section discusses my recommendations for further improvements to the system.

A third chapter, beginning on page 75, discusses the methods used in the investigation in greater detail. The flow of the paper, therefore, will follow Figure 7 below.

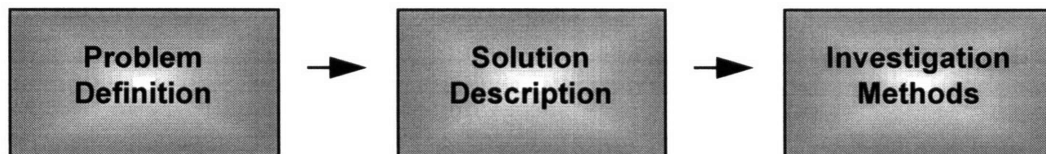


Figure 7. Description of the flow of this document.

II. CONCLUSIONS AND RECOMMENDATIONS

This chapter is divided into three main sections. The first section is designed to describe the new and redefined metrics that I developed to allow key groups of customers (technicians, engineers, and managers) to gauge the performance of the manufacturing area and to pinpoint problems. The second section is designed to provide an overview of the tracking and reporting system that I developed to deliver the new metrics to the key customer groups. The last section outlines a few recommendations for further investigation.

PERFORMANCE METRICS

There were three yield numbers that were widely used as process metrics in the Component Manufacturing Area. They were the First-Pass Yield, the N-Pass Yield, and the Terminate Yield. Monthly reports of N-Pass Yields and Terminate Yields were provided by manual and semi-automatic methods to key members of the manufacturing area, while the calculation of First-Pass Yields was primarily restricted to the individual technicians' logbooks. One of the goals of my project was to provide yield numbers automatically.

In order to generate these numbers with a computer program, I wanted to make sure that everyone shared a common interpretation of their definitions. This

was especially important in unique cases relating to rework, and in cases where there were significant time delays. (For example, if a unit starts work at Operation 10 on the last Friday in May, goes on "Hold," and is scrapped at the operation on the first Monday in June, does the terminate get "charged" against the yield for *May* or *June*?) Also, I wanted to be sure that generating these metrics was indeed necessary.

Otherwise, the effort involved in collecting the data to do so would be wasted. Lastly, I wanted to determine exactly what data was needed to successfully calculate the metrics. Once the minimum data requirements were determined, I could suggest ways to eliminate the collection of extraneous data.

In addition, there were two *new* metrics that were developed. They were the *Root Cause Yield* and the *Average Number of Passes*. The new metrics were easy to generate with a computer, and they grew out of input and feedback from Kate Stohlman, the Component Manufacturing Section Manager, Stu Siegel, the Component Manufacturing Engineering Manager, and Martha Keller, the Component Assembly Production Supervisor.

This section of Chapter Two is intended to define the five metrics (three new ones and two old ones) that were used in the yield reporting system that I built. I will also describe why I chose these particular metrics as being important enough to track.

First-Pass Yield

The first metric that I will tackle is First-Pass Yield. To begin, I will explain

what it is, then I will try to outline reasons why I felt that it was important enough to measure.

What is it?

For a single operation, First-Pass Yield refers to the outcome of a first run-through of fresh⁷ units. The figure is composed of a numerator and a denominator such that:

$$\text{First-Pass Yield} = \frac{\text{numerator}}{\text{denominator}} = \frac{\text{all parts that were good the first time through}}{\text{units started that day}}$$

Successfully reworked units are not counted as "good" in this calculation. Similarly, units that are initially good, but are later scrapped when they are returned to be reworked, are not counted as "bad" in the calculation. (Please see Appendix C on page 103 for an example.) In effect, the First-Pass Yield is saying, "Out of X new units, we successfully completed Y on the first try." In such a case, Y/X would be the First Pass Yield. Below are examples of what the numerator and denominator components of this number represent.

Numerator: The number of successes resulting from the first time an operation is performed.

includes:

- Good Units

⁷"Fresh" indicates that it is the first time that the unit has ever passed through the operation. A unit that is being reworked at an operation is no longer considered to be "fresh."

Numerator (cont...)

- Units put on "Hold", but later determined to be acceptable

does not include:

- Units at an operation from a previous day that are successfully reworked
- Units that failed, but were successfully reworked at the operation
- Scrapped units

Denominator: The number of starts for the first run-through for an operation on a particular date

includes:

- All fresh units started at an operation for a given date

does not include:

- Units at an operation that are on "Hold" from any previous days
- Any reworks from other operations
- Units held over for unfinished rework from any previous day

Why measure it?

First-Pass Yield is a good indicator of process robustness. A high First-Pass Yield means that the process is good at overcoming the variation in incoming materials. It may also signify that the technicians aren't making very many mistakes. Conversely, a low number can point to problems with component parts or with the process itself. Further, a string of low First-Pass Yields is an immediate indication of trouble in a process. In many ways, it is the first sign that a process is not performing up to expectations.

N-Pass Yield

Like First-Pass Yield, N-Pass Yields were regularly monitored and reported when I arrived at the division. However, there wasn't a common definition of its meaning or a consensus of the proper method of calculation. Below is the definition that I used to build the reporting system.

What is it?

For a particular operation, the N-Pass Yield reflects the percentage of all units, started at a particular operation, that were passed on to the next operation without being scrapped. The figure is composed of a numerator and a denominator such that:

$$N\text{-Pass Yield} = \frac{\text{numerator}}{\text{denominator}} = \frac{\text{units started that day} - \text{all units scrapped at that operation}}{\text{units started that day}}$$

In effect, the N-Pass Yield number is saying, "Out of X new units, Y eventually were successfully made available for work at the next operation." In such a case, Y/X would be the N-Pass Yield. Likewise, for an entire manufacturing area (i.e. Component Manufacturing), the N-Pass Yield is the percentage of starts are shipped to the next manufacturing area (i.e. Wire and Test). Note that units scrapped downstream are not "charged back" to the operation associated with the failing defect code, but are charged against the yield for the operation where they failed.

Numerator: The number of units which eventually successfully exit a particular operation

includes:

- Good units
- Successful reworks of units started that day
- Units that go on "Hold" and are later determined to be acceptable

does not include:

- Units held over for unfinished rework from any previous day
- Units at an operation that are on "Hold" from any previous days

Denominator: The number of units started at a particular operation on a particular day

includes:

- All fresh units started at an operation for a given date

does not include:

- Units at an operation that are on "Hold" from any previous days
- Any reworks from other operations
- Units held over for unfinished rework from any previous day

Why measure it?

The N-Pass Yield number paints a picture of process capacity. A high N-Pass Yield indicates that a high percentage of the units passing through the process will eventually be shipped to the customer. While a low *First-Pass Yield* number is a sign that many units may need to be reworked at an operation before they are shipped, a low *N-Pass Yield* is a further indication that even rework efforts at an operation are not successful.

Whether high or low, a *consistent* N-Pass Yield helps managers and

supervisors to forecast production capacity or output, and perform production scheduling.

Terminate Yield

The third and final metric in use at the time this investigation started was the Terminate Yield. There was some confusion among various members of the Component Manufacturing Area between this metric and the N-Pass Yield. Recall from the introduction to this report that the Terminate Yields represented in the *semi-automatic report* were sometimes mistakenly used interchangeably with the N-Pass Yields represented in the *manual report*. I feel it necessary to define the term so that it could be distinguished from the N-Pass Yield.

What is it?

For a particular operation, the Terminate Yield reflects the percentage of units that leave an operation successfully during a time period. It is easier to calculate and produce because it doesn't take into account the date and time that units enter the operation (or "start"). For this reason, it can be applied to an entire manufacturing area more easily than any of the other metrics described in this section. The time lag introduced between operations is immaterial to the calculation, and the problems associated with the calculation of a Composite N-Pass Yield are avoided. (See the discussion on Composite Yields on page 22.)

The figure is composed of a numerator and a denominator such that:

$$\text{Terminate Yield} = \frac{\text{numerator}}{\text{denominator}} = \frac{\text{total number of terminates} - \text{number of scrapped units}}{\text{total number of terminates}}$$

In effect, the Terminate Yield number is saying, "Out of X units that left the operation (for any reason) during the specified time period, Y were successfully shipped as good units to the next operation." In such a case, Y/X would be the Terminate Yield. Likewise, for an entire manufacturing area (i.e. Component Manufacturing), the Terminate Yield is the percentage of all units that left the manufacturing area as good units and were shipped to the next manufacturing area (i.e. Wire and Test). Note that downstream terminates are not "charged back" to the operation associated with the failing defect code.

Numerator: The number of units which eventually successfully exit a particular operation

includes:

- Good units
- Successful reworks
- Units that go on "Hold" and are later determined to be acceptable

does not include:

- Units held over for unfinished rework from any previous day
- Units at an operation that are on "Hold" from any previous days

Denominator: The number of units leaving a particular operation on a particular day

includes:

- All terminates, whether good units or scrap

does not include:

- Units at an operation that are on "Hold" from any previous days
- Any reworks from other operations
- Units held over for unfinished rework from any previous day

Why Measure It?

Since the practice of calculating an accurate Composite N-Pass Yield is so difficult, the Terminate Yield is the only realistic alternative metric for establishing an overall process yield. However, there two key difficulties inherent in this number:

- 1.) The number is not a true "yield" number in that it doesn't account for the number of "starts" into a process in a particular time period. It only measures "finishes."
- 2.) Because the throughput time of a component is long, many of the units accounted for in any given month's Terminate Yield may have entered the process the month before. This introduces a time lag into the metric. For this reason, its usefulness is limited to some higher multiple of the throughput time. In other words, it loses its meaning in yield calculations that are representative of times shorter than one month.

Because of these limitations, I was very hesitant to include this metric in the yield reporting system designed as part of this project. I will caution that its only real usefulness is as an alternative to the traditional calculation of Composite N-Pass Yield. While, as a metric, it isn't as good a as the other four discussed here, it is somewhat more accurate for reporting actual process performance than the Composite

Yield used in the *manual report*.

Root Cause Yield

In addition to the three metrics in place at the division before the project began, I found it necessary in the course of my investigation to develop two new performance measures to include in the tracking and reporting system. With the help of several key players in the division, the first one I developed is called *Root Cause Yield*.

Root Cause Yield grew out of the problem of assigning responsibility for defects. Technicians at Operation 10, or some other early operation, desired a formal method of knowing whether or not the units that they built on a given day were ultimately shipped to customers as "good" units. Alternatively, technicians at any operation desired a way of knowing the quantity and nature of defects discovered at downstream operations that might have inadvertently been introduced in their own processes. Similarly, technicians at downstream operations were often "charged" for defects that were actually due to problems at upstream operations, resulting in the lowering of their yields. The Root Cause Yield is part of the solution to these problems.

What is it?

The Root Cause Yield represents the percentage of units from a particular

operation that were scrapped, at that operation *or* a downstream operation, for a defect code associated with that operation. The figure is composed of a numerator and a denominator such that:

$$R\ C\ Yield = \frac{\text{numerator}}{\text{denominator}} = \frac{\text{units started that day} - \frac{\text{all units scrapped at that operation for defects associated with that operation}}{\text{units started that day}} - \frac{\text{all units scrapped at downstream operations for defects associated with that operation}}{\text{units started that day}}}{\text{units started that day}}$$

In effect, the Root Cause Yield number is saying, "Out of X new units started at my operation today, Y eventually went on to be shipped to the Wire and Test Area without being scrapped by someone for a defect related to my operation". In such a case, Y/X would be the Root Cause Yield.

Numerator: The number of units that eventually successfully get shipped to the next manufacturing area without being scrapped for a defect code related to the particular operation

includes:

- Good units
- Successful reworks of parts started that day
- Units that go on "Hold" and are later determined to be acceptable
- Units that are scrapped at the operation for defects that are related to a different operation

does not include:

- Units held over for unfinished rework from any previous day
- Units at an operation that are on "Hold" from any previous days
- Units that are scrapped at an operation for generic defects.

Denominator: The number of units started at an operation on a particular day

includes:

- All fresh units started at an operation for a given date

Denominator (continued...) does not include:

- Units at an operation that are on "Hold" from any previous days
- Any reworks from other operations
- Units held over for unfinished rework from any previous day

Why measure it?

Today, the average technician has no immediate method for determining whether his or her units are making it through the whole process successfully. Most technicians *are* informed when a large number of units that they built fail at downstream operations. However, the communication channels are only informal. When surveyed, a number of the technicians commented that they didn't frequently get feedback on downstream defects until the number of occurrences was at crisis stage, and that they would rather get earlier, regular feedback.

Also, this metric is a true measure of process robustness. For example, if products leave Operation 10 successfully, only to be scrapped the next day for a defect related to Operation 10, Operation 10 isn't really successful. While scrapped units contributing to N-Pass Yields are not "charged-back" to the originating process, scrapped units contributing to Root Cause Yield *are* charged back. Taken in combination, the N-Pass and Root Cause Yield metrics provide a lot of information.

Average Number of Passes

Along with Root Cause Yield, the Average Number of Passes that a unit needs

to successfully complete an operation is a new metric that I chose to define and include in the system. The need for it grew out of the fact that some units were believed to be living in "infinite rework loops" at particular operations. As such, there needed to be some method for capturing this information. The figure is composed of a numerator and a denominator such that:

$$\text{Average Number of Passes} = \frac{\text{numerator}}{\text{denominator}} = \frac{\text{total number of passes a unit(s) makes through an operation before being terminated}}{\text{units started that day}}$$

What is it?

This metric refers to the number of successive attempts that are required for a unit to complete an operation. To illustrate, assume that two units are started on a particular day. One unit takes three tries to pass the operation, while the other is deemed acceptable on the first try. The Average Number of Passes, therefore, is two.

Numerator: The number of times a unit (or units) passes through an operation before successfully being completed or scrapped

includes:

- All first-time passes, whether resulting in a rework event, good part, or scrap event
- Each successive rework attempt, whether it happens the same date or a subsequent date, and whether the result is a another rework event, good part, or scrap event

does not include:

- Units at an operation that are on "Hold" from any previous days
- Any reworks from other operations

Numerator (cont...) ● Units held over for unfinished rework from any previous day

Denominator: The number of complete units from an operation, including rework units, for a particular date

includes:

- Good units and scrapped units
- Successfully reworked units
- Units that go on "Hold" and are later determined to be acceptable

does not include:

- Units held over for unfinished rework from any previous day
- Units at an operation that are on "Hold" from any previous days

Why measure it?

This metric provides quantitative information as to the success rate of rework at a particular process. There had been talk for some time in the Component Manufacturing Area that more attention should be paid to the tracking of rework at each process, and this metric is a good way to discover how well the process is handling rework loops. It can also help to determine how much raw material to order for the area. For example, a process may ship ten good units per day, but use 4 lbs. of material to accomplish the task, because each unit had to be reworked 10 times.

Additionally, there were a few processes in the system that seemed to have more rework than others. As engineers worked to reduce the rework at these operations, they needed a method to track their success. By studying changes in the Average Number of Passes for a particular operation over time, an engineer could gauge the success of his or her work.

Summary of Performance Metrics

No single metric mentioned here, taken alone, is adequate to paint a thorough picture of the performance of a production line by itself. Instead, all five are necessary to give a complete description of the performance of an operation, or of the entire manufacturing area.

Taken alone, First Pass Yield cannot give engineers, managers, and technicians a good illustration of the throughput potential of the production line. The calculation of N-Pass Yield is necessary to provide this piece. But N-Pass Yield cannot necessarily assign the cause of a particular defect to a specific operation. To do this, the calculation of Root Cause Yield is helpful. It is important to understand that all of the metrics presented here are useful in their own way, but none are powerful enough to represent the performance of an entire manufacturing area.

It may be helpful for the reader to understand these metrics by studying an example. An example of the application of these metrics to a sample scenario may be found in Appendix C on page 104.

THE YIELD AND DEFECT REPORTING SYSTEM

The main element of my work at Hewlett-Packard was the development of an automatic yield and defect tracking and reporting system. In this system, the line technician uses a bar-code reader to enter the unit number of each unit that enters and

exits his or her operation. He or she also enters the operation number corresponding to the task at hand. This information is then fed into a database comprised of several different tables, with each table fulfilling a different function. Later, when an technician, engineer, or manager wishes to look at the yield for an operation, locate a unit on the floor, get a breakdown of the defects for a particular family of part numbers, or download the data into a spreadsheet program, information can be pulled from the database via a user-friendly application program.

In this section of Chapter Two, I will outline the workings of this system. First, I plan to explain the appropriateness of the level of automation that I built into the system. I want to explain why bar-coding every unit at every operation is a fitting solution to the situation at hand. Then, I will describe the basic infrastructure of the system. Next, I intend to explain the database architecture that the system uses. This will not be in too much detail, but I feel that an overview will help the reader understand how to set up similar systems. Finally, there will be a section explaining the PC-based reporting application that I developed.

The Nature of the Data Collection Scheme

As I mentioned in an earlier section of this report, this organization was using a manual collection method to gather data on the manufacturing floor. In many ways, this system was entirely adequate and suitable to the task. Indeed, the system had evolved over several years through thoughtful revisions and careful planning.

However, as the complexity of the parts being produced by the floor increased, and market demand drove up product mix and volumes, the adequacy of the system diminished, and it became clear that revisions were necessary. In re-engineering the data collection scheme, I had to answer two questions. First, which type of collection and reporting scheme should be used (namely, would a manual, semi-automatic, or fully automatic system be most appropriate), and second, whether a commercial system from an outside vendor should be purchased, or an in-house system developed. The answers to these questions are found in the following subsections.

Collection and Reporting Schemes

In any manufacturing environment, there has to be a suitability of the level of automation to the task at hand. One popular graph used to explain this is shown in Figure 8 on page 55. The graph represents a rather traditional view of the "manufacturing hierarchy," and it can be used as a tool for managers to decide what level of automation to use for a particular situation. For small quantities of highly specialized parts, stand-alone NC machines or manual assembly is appropriate. As volumes increase and product variety decreases, installation of highly automated, fixed transfer lines may be warranted. This attention to "fit" is the subject of many operations management courses at MIT.

Equally important, however, is the fit of information systems that govern manufacturing processes to each unique situation. The investment involved in setting up an information system, whether it be for tracking defects or managing inventory,

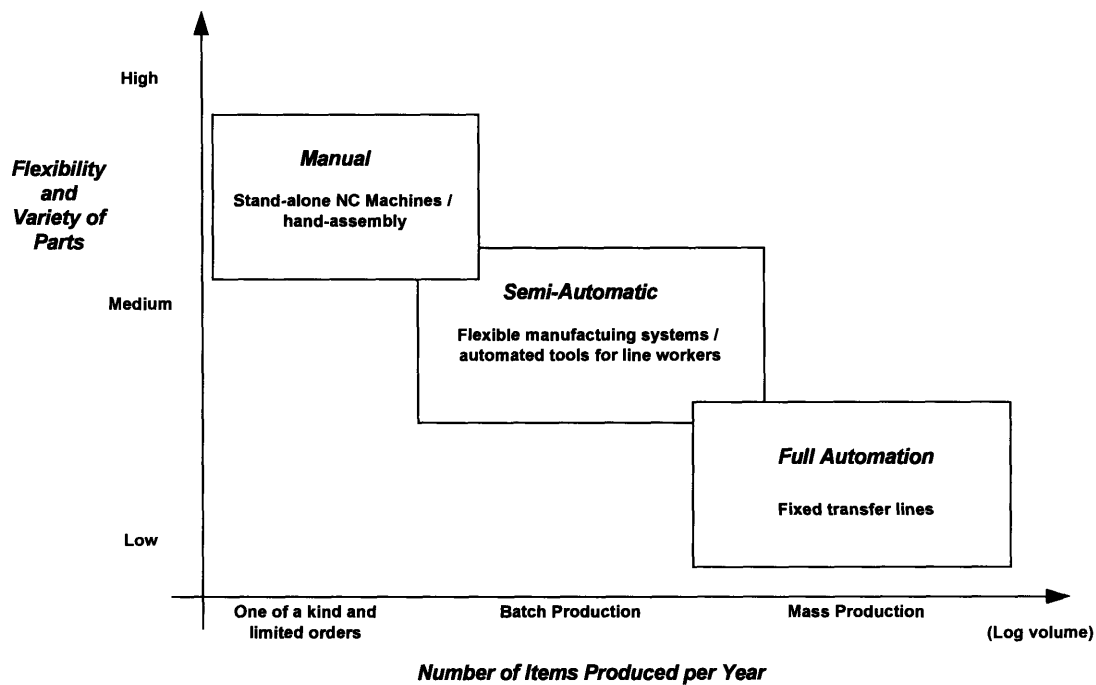


Figure 8. Manufacturing Hierarchy. Adapted from Nahmias, Steven, *Production and Operations Analysis*, (Boston: Irwin, ©1989), page 614.

can be huge. Many companies are struggling with justifying the costs of these systems. In the 1980's, some large firms dumped millions of dollars into highly automated MRP systems that didn't always work as well as the few dollars spent by other companies on a handful of Kanban cards and magic markers. It isn't that the Kanban cards were necessarily better or worse than the mainframes required to run the MRP systems (the debate of which is a thesis paper unto itself); the point is that companies need to pay attention to the suitability-to-task of the systems that they choose to employ. More simply, it is important for companies to "use the right tools for the right job."

Manufacturing engineers faced with the task of implementing production performance systems have a variety of tools available to them, from simple SPC

charts to complex automatic control systems housed in central computers. To help these engineers (and myself) make decisions about which tool to use, I took the chart in Figure 8 a step further. Placing the *level of automation* on the y-axis and the *production rate* on the x-axis, I constructed the chart shown in Figure 9 below. To use the matrix, the user merely finds the organization's place in the graph. In this division's case, different parts of the manufacturing organization fit into different

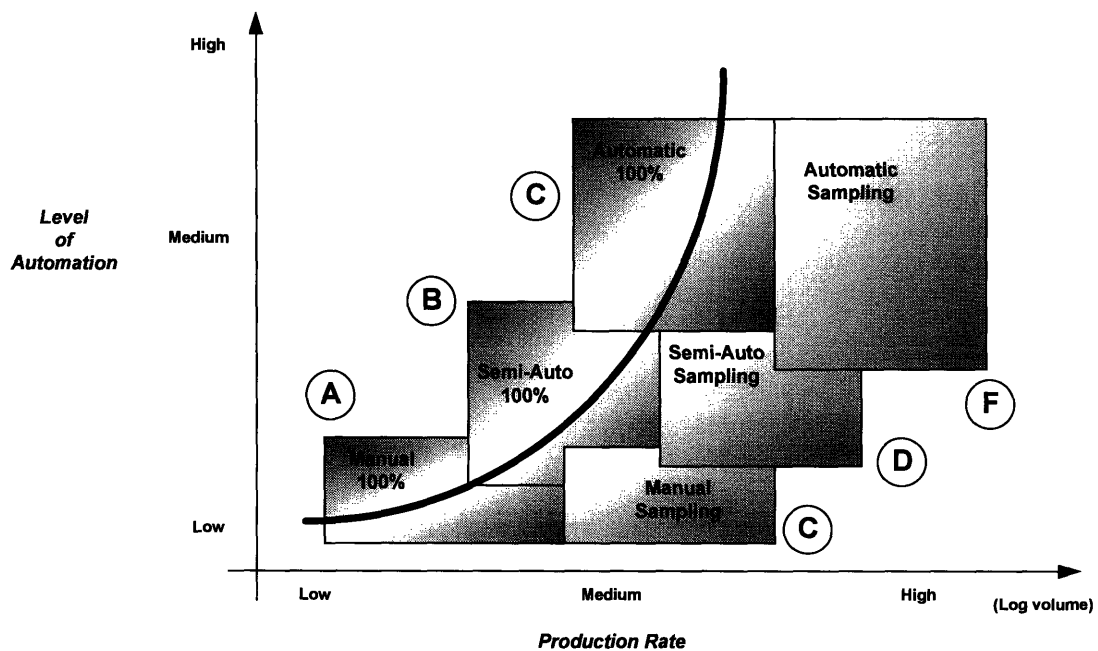


Figure 9. Manufacturing Information Strategy Matrix.

parts of the matrix. The Component Assembly portion of the manufacturing area had traditionally fit squarely in Box A. By the time this investigation had started however, it had moved up the curved line into Box B.

In order to better understand what is meant by "Semi-Auto Sampling," and the other terms used in Figure 9, the reader should refer to Table III on the next page. In

this table are explanations for each type of strategy, along with examples of the types of manufacturing organizations that might find such strategies useful.

Table III
Explanation of Information Strategy Matrix

Position in Matrix	Data Collection Frequency	Data Collection Method	Sample Strategy	Sample Organization
A	100%	Manual	Human data collection in logbooks or journals. Notes on every unit built. Infrequent summary reports.	Custom built surfboards or other craft assembly work.
B	100%	Semi-Automatic	Computer assisted data collection. Computer generated reports on an as-needed basis.	Electronic component assembly or other low volume, high tech product requiring data for every unit produced.
C	100%	Automatic	Automatic data collection by robots or other equipment in the production process. Automatic, computer generated reports.	Two-way radio or paging product production. Higher volume products built on heavily automated lines where information on each product aids in customer order fulfillment. (e.g. particular radio frequencies.)

Position in Matrix	Data Collection Frequency	Data Collection Method	Sample Strategy	Sample Organization
D	Statistical Sampling	Manual	Human data collection on SPC charts. Infrequent reports on production runs required for planning purposes.	Lighting fixture assembly or other commodity product generally built by hand in low labor-cost countries.
E	Statistical Sampling	Semi-Automatic	Computer assisted data collection or SPC. Production reports generated by computer or with computer assistance.	Automobile assembly or other assembly line process.
F	Statistical Sampling	Automatic	Automatic data collection by robots or other equipment in the production process. Automatic, computer generated reports. Feedback control systems.	Automobile parts assembly on high volume lines or other products where complex testing procedures and high volumes preclude 100% testing.

In addition to the reasons outlined in the matrix and the table above, there was other motivation for going with "Strategy B" information system in the Component Manufacturing Area. First, the FDA requires the collection of a wide variety of data on 100% of the units produced on the line. This is not uncommon in the medical products industry. Second, the technicians (i.e. customers of the system) *wanted* to

use computers to collect and retrieve data from their areas. This was evident in my conversations with them, and through the results of a survey conducted in the area. For more information on the survey and results, see the section titled "Technician Yield Reporting Survey" on page 80.

Outside vs. In-House Development

Once the strategy was determined, namely a semi-automatic collection and reporting scheme using computers to do the report generation, a sourcing decision had to be made.⁸ Several full-service systems are available on the market for accomplishing the goals of the project. Indeed, HP itself makes highly regarded factory floor management products that it sells on the outside world. By purchasing one of these systems, Components Manufacturing and I could save months of time developing and coding an application. However, there also were several drawbacks to pursuing this strategy.

First, commercially available systems are expensive, commonly costing upward of \$30,000 for the software alone. HP was not necessarily ready to put resources toward a new information system when the one that they were using might still be adequate. The reader should keep in mind that this project was just one building block in the long-term strategy for factory floor information systems at the division (see Figure 6 on page 31). Implementing a full blown OEM system at this

⁸In actuality, the two decisions were made very closely together. Since one impacts the other so greatly, it was necessary to keep the impact of one decision in mind when deciding on the other.

stage of the game might slow progress toward subsequent stages. Also, HP had very talented people in-house. Much of the needed expertise could be obtained by looking inside at a fraction of the cost of outside consultants.

Second, HP had relied on an outside vendor for the existing QIC System. QIC had been a relative success, but making modifications in-house was impossible, due to the fact that the vendor "owned" the code for the system. The division was operating in a business full of change, where new products were being developed regularly. As new needs arose, they could not easily or quickly modify the QIC System to adapt to these changing needs. This was a point of continuous frustration for several members of the manufacturing engineering staff. By developing a new system in-house, HP would have the ability to alter the system to adapt to the changing business environment.

Third, the company was, in many ways, a little unsure about what its needs would be for the future. By developing a small system in-house, they could understand how to best use computers to track information on the manufacturing floor. As such, this project could be a learning tool. By studying the good and bad points of this project, the division could be better prepared to communicate its needs to an outside vendor, should it ever consider going outside in the future for other projects.

Database Infrastructure

In this section, I will explain the database system infrastructure used for the yield and defect tracking and reporting system. Figure 10, shown below, is a diagram of the system. Bar-code readers at each operation are managed by a PC running a transaction processing program. As the bar-code readers feed data into this Transaction Manager, the validity of the data is checked (i.e. "Is the unit entering Operation C before completing Operation B?"). If the data is invalid, the Transaction

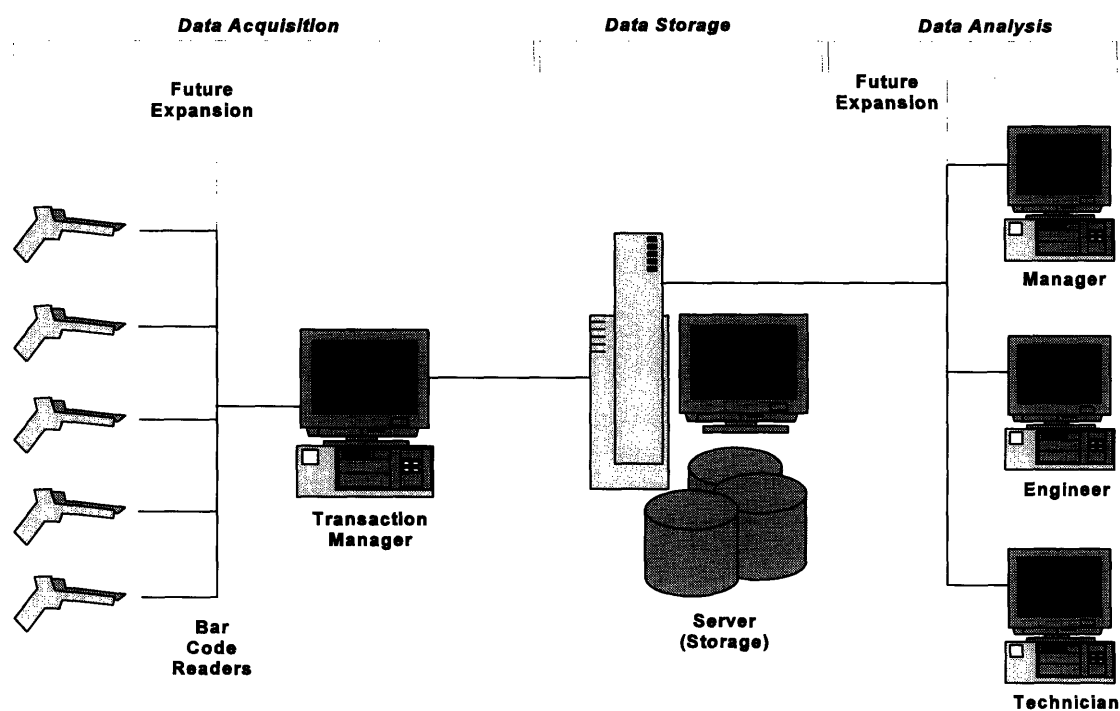


Figure 10. Diagram of the infrastructure of the information system. Note that it is fully expandable in the acquisition, storage, and analysis sections.

Manager notifies the line technician via a readout on the bar code terminal. If the information passes the validity checks, however, it is passed to the server for storage.

When information is needed, PCs running application software search for the raw data necessary to calculate yields and produce defect Pareto Charts. (This method puts the actual computation burden in the hands of the local PC.) This client-server approach to computing helps the system to be easily expandable. As the needs of the system grow, the users can add more PCs. Central storage capacity may have to be added, but central processing power, which is expensive, does not.

Database Architecture

Database tables are lists of information organized in rows and columns. The rows are known as *records*, while the columns are referred to as *fields*. In *relational databases*, certain fields in one table are linked to fields in other tables.

In this application, the database that stores the information entered via the barcode readers has four main tables, and is of the relational variety. Figure 11 on the next page shows the four tables and the field names of the stored information. For example, the *HISTORY* table contains information on a particular unit's Part Number, Issue Date, Issue Time, etc. The asterisks next to some of the field names indicate *keyed fields*. In each table, no two records have the same information in the keyed fields. Also note that the tables can be linked via the Unit Number. As such, all units passing through the Component Manufacturing Area have unique Unit Numbers. See Table IV on the page 64 for a sample record from the *HISTORY* table.

Each of the four main tables serves a particular function. In the next

HISTORY	ACTIVITY	UNITS
UNIT_NUMBER* PRODUCT PART_NUMBER SOURCE SOURCE_REF SERIAL_NUMBER STATUS ISSUE_DATE ISSUE_TIME FINISH_DATE FINISH_TIME AREA_SHIFTS GROSS_CYCLE NET_CYCLE GROSS_SHIFT NET_SHIFT GROSS_HOLD NET_HOLD CURRENT_OPERATION SCRAP_DEFECT SCRAP_CAUSE SCRAP_COMMENT SCRAPPED_BY DEFECT_COUNT ACTIVITY_COUNT	UNIT_NUMBER* STATUS* OPERATION* ACTIVITY_NUMBER* PART_NUMBER GROSS_CYCLE NET_CYCLE GROSS_SHIFT NET_SHIFT START_DATE START_TIME FINISH_DATE FINISH_TIME OP_SHIFTS HOLD_CODE	UNIT_NUMBER* PRODUCT_TYPE PART_NUMBER SOURCE SOURCE_REF SERIAL_NUMBER STATUS ISSUE_DATE ISSUE_TIME AREA_SHIFTS GROSS_CYCLE NET_CYCLE GROSS_SHIFT NET_SHIFT GROSS_HOLD NET_HOLD CURRENT_OPERATION RECENT_ROUTE_OP OP_START_DATE OP_START_TIME OP_SHIFTS OP_GROSS_HOLD OP_NET_HOLD HOLD_CODE HOLD_START_DATE HOLD_START_TIME HOLD_COMMENT DEFECT_COUNT ACTIVITY_COUNT REWORK_OPERATION
	DEFECTS UNIT_NUMBER* DEFECT_NUMBER* STATUS PART_NUMBER DATE TIME DEFECT_CODE CAUSE_CODE OPERATION DEFECT_COMP_PART_NBR DEFECT_COMP_SERIAL_NBR DEFECT_LOCATION_REF DEFECT_COMMENT SCRAPPED_BY ANALYZED	

Figure 11. The four main tables in the yield and defect reporting system. The asterisks indicate the keyed fields.

subsections, I will try to explain the function that each table serves.

The UNITS Table

The *UNITS* table acts as a place-holder in the database for each unit. Each unit in the manufacturing area has one record in the *UNITS* table. When the unit enters the manufacturing area, a technician enters biographical information about the unit. By biographical information, I'm referring to the unit's Part Number,

Table IV
Sample Record from the History Table

unit_number	product	part_number	source	source_ref
A5000	SENS	21246-64200	New	0
serial_number	status	issue_date	issue_time	finish_date
1234567	COMPLETE	01/01/93	0.3320313	01/02/93
finish_time	area_shifts	gross_cycle	net_cycle	gross_shift
0.6054688	0	1.273438	0.5585938	0
net_shift	gross_hold	net_hold	current_op	scrap_defect
0	0.7148438	0.7148438	90	
scrap_cause	scrap_commnt	scrapped_by	defect_count	activity_count
			0	7

Source, Reference Numbers, etc. Further, as the unit enters and exits each operation, the *UNITS* table is updated with the current operation number. The table also holds information pertaining to the time the unit spends being "worked-on" at each operation, time spent "on-hold" for engineering evaluation, and information about the total time that the unit has spent in the manufacturing area. Each unit only has one record in the *UNITS* table at any given time. Once the unit has completed its journey through the manufacturing area, the record is deleted. As such, the *UNITS* table is a listing of the location of every unit on the manufacturing floor at any given time.

The *ACTIVITY* Table

The *ACTIVITY* table keeps a record of every activity that each unit performs. When a unit successfully completes an operation, a record is entered into the *ACTIVITY* table indicating so. Similarly, rework events, repair steps, hold events, and scrap incidences are also recorded in their own separate records. The table also keeps a count of how many activities that the unit experiences. By examining the activity table, one can find a complete "life-story" of each unit that has passed through the manufacturing area. Typically, there will be about a dozen records in the *ACTIVITY* table for each unit.

The *DEFECTS* Table

The *DEFECTS* table, as the name implies, keeps a running tab of all the defects found for each unit on the manufacturing floor that result in rework events or scrap actions. A Defect Code and Cause Code are entered by the line technician when a defect is found. A typical defect code might be "3004," which may stand for "Unit Shorts Out During Electrical Test." Initially, the values for Defect Code and Cause Code are the same. However, the Cause Code is often changed after the defective unit is torn apart and inspected. In the above example, the Cause Code might be "3120," which may signify "Shorts Out Due to Faulty Wires."

The *HISTORY* Table

This table keeps a running list of all the units that have passed through the

manufacturing floor, and each unit has only one record in the table. A particular unit's record tells how long it took to build the unit, where the unit came from, whether or not it was a prototype, and other information.

Use of the System - Entering Data

As each unit enters and exits an operation, the technicians scan the unit with a bar-code reader and execute operations on the bar-code terminal. For example, when a unit enters an operation, the START command is entered at the terminal. This action causes the UNITS table to be updated with the unit's exact location. After the line technician finishes work on the unit and successfully tests it, the FINISH command is entered, signifying that work is complete. This action causes a record to be written to the *ACTIVITY* table, signifying that the unit has completed yet another activity. Figure 12 and Figure 13 on pages 68 and 69 give a lengthy example of what happens during the production of a toy car. For this example, assume that the toy car only has to go through three operations to be successfully assembled: Operation 10, Operation 20, and Operation 30.

Other Tables

In addition to the four main tables listed above, there are other tables that were developed to help the program do its job. One table, the *OPERATE* table, contains a list of each part number and the corresponding operations that are required to successfully build it. This allows the program to check the progress of each unit, and

to ensure that each unit passes through Step A before someone tries to enter it into Step B. The *DFCT_CDE* table lists all of the legal defect codes that a line technician can choose from to describe product defects. It also contains information that links particular defects with the operations that are normally associated with creating each defect. For example, if bumpers are attached at Operation 10, the defect "Bumper Fell Off" would be linked to Operation 10 in the *DFCT_CDE* table. This table is used by the database application to calculate Root Cause Yield.

The accuracy of this table is key to the success of the system. If any of the information in the *DFCT_CDE* table is incorrect, the resulting Root Cause Yield calculations will be faulty. Care must be taken to ensure that this defect code "map" is kept current.

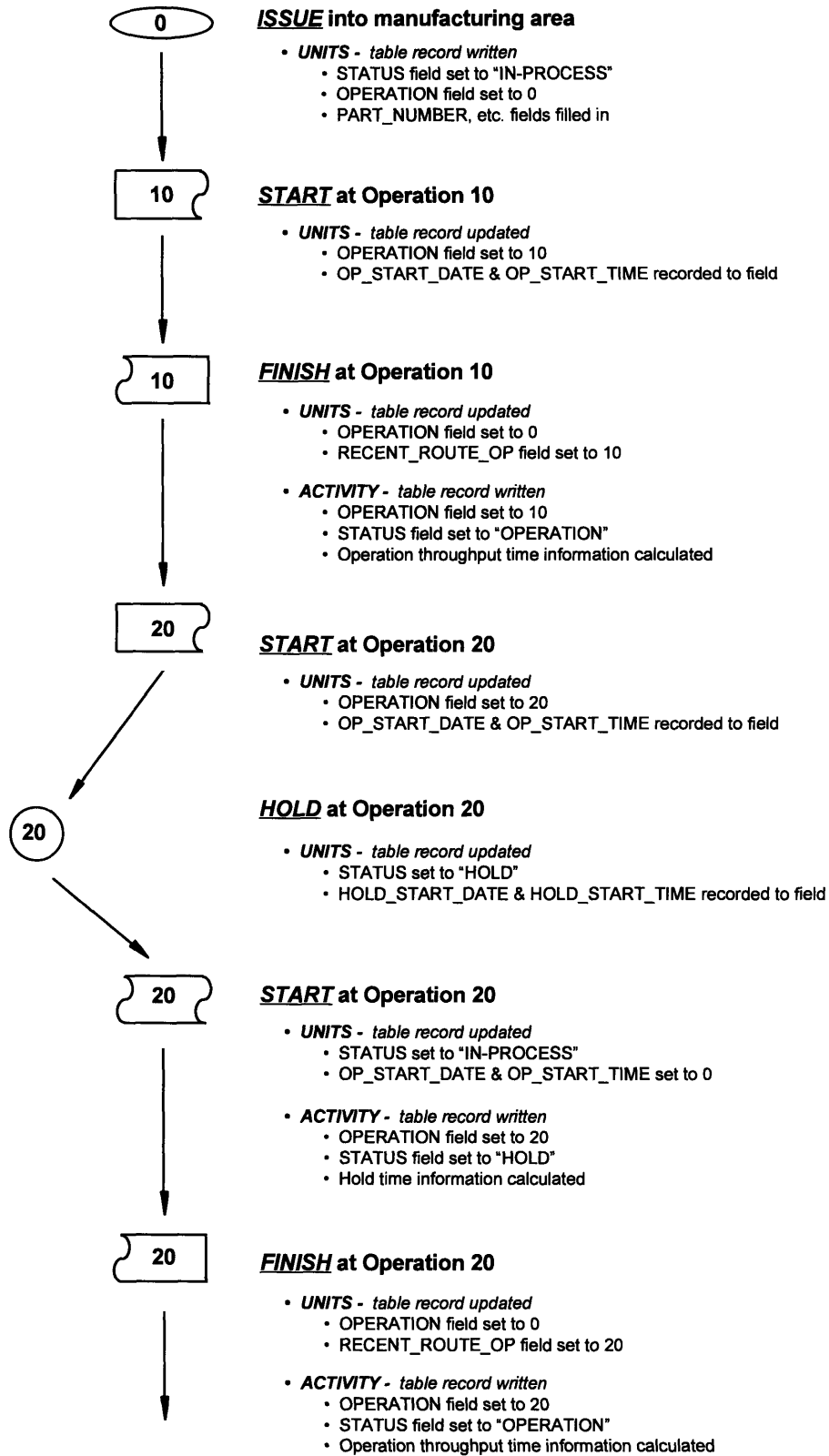
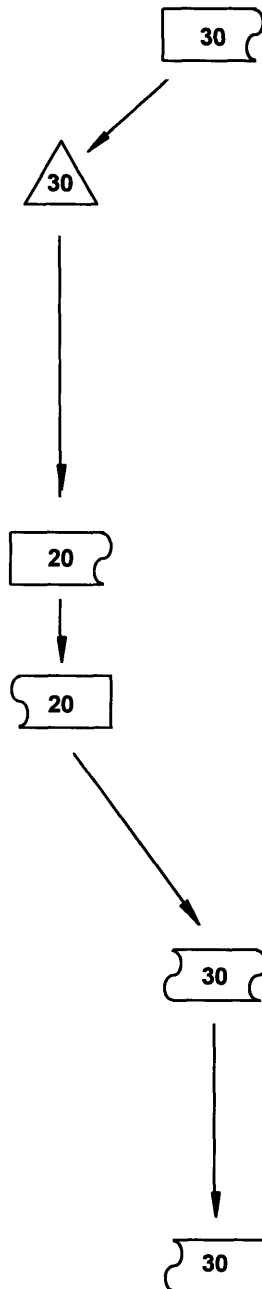


Figure 12. Progress of a unit through the manufacturing process. The underlined words are actions entered into the bar code terminals by line technicians



START at Operation 30

- **UNITS** - table record updated
 - OPERATION field set to 30
 - OP_START_DATE & OP_START_TIME recorded to field

REWORK at Operation 10

- **UNITS** - table record updated
 - STATUS set to "REWORK"
 - REWORK_OPERATION set to 30
- **ACTIVITY** - table record written
 - OPERATION field set to 30
 - STATUS field set to "INCOMPLETE"
- **DEFECTS** - table record written
 - OPERATION field set to 30
 - STATUS field set to "REWORK"
 - DEFECT_CODE field information recorded
 - DATE and TIME fields recorded

START at Operation 20

- **UNITS** - table record updated
 - OPERATION field set to 20
 - OP_START_DATE & OP_START_TIME recorded to field

FINISH at Operation 20

- **UNITS** - table record updated
 - OPERATION field set to 0
 - RECENT_ROUTE_OP field set to 20
- **ACTIVITY** - table record written
 - OPERATION field set to 20
 - STATUS field set to "REWORK"
 - Operation throughput time information calculated and recorded

START at Operation 30

- **UNITS** - table record updated
 - OPERATION field set to 30
 - REWORK_OPERATION set to 0
 - STATUS field set to "IN-PROCESS"
- **ACTIVITY** - table record written
 - OPERATION field set to 30
 - STATUS field set to "REPAIRED"
 - Rework loop throughput time calculated and recorded

FINISH at Operation 30

- **UNITS** - table record deleted
- **ACTIVITY** - table record written
 - OPERATION field set to 30
 - STATUS field set to "OPERATION"
 - Operation throughput time information calculated and recorded
- **ACTIVITY** - table record written
 - OPERATION field set to 30
 - STATUS field set to "COMPLETE"
 - Process throughput time information calculated and recorded
- **HISTORY** - table record written
 - UNIT_NUMBER recorded
 - STATUS field set to "COMPLETE"
 - Process throughput time information calculated and recorded

Figure 13. Second half of the sample unit's progress through the manufacturing area.

The Yield Reporting and Defect Tracking Application

The main portion of the project was the construction of a database application program to track defects and report yields. In the following section of Chapter Two, I will explain the application and its function.

When units are scanned on the production floor, data is inserted into a database. To look at this data in an intelligent manner, an application program was written. Contained in the application program is software code for PCs. (PCs could be found on the engineers' desks, the managers' desks, and on the production line in the Components Manufacturing Area.) This code searches the database for relevant information based on the user's requests, calculates the yield numbers required by the user, then displays the results on the screen. At the user's option, the application will also produce a hardcopy yield report, or export raw data to a spreadsheet file. The application that I built runs under Microsoft Windows, an operating system commonly used at HP, and familiar to the employees at the division.

The database engine used to store the information is Paradox. Paradox is in fairly wide use at the division, and it is the system upon which QIC was based. (For more information on why I chose to continue using Paradox, see page 84.)

The application was written in ObjectPAL code. ObjectPAL is an object-oriented programming language that is shipped with Paradox for Windows. It has its roots in C, and it shares syntax with C++. The language is very strong, making it more suitable for building full-service network applications than Microsoft

QuickBasic, the language shipped with Microsoft's Access product, or the macro language contained in Lotus' Approach product.

I decided to name the application MARS, for Manufacturing Area Reporting System. Since it is a Windows application, the user gains access to it via the Windows Program Manager screen. Figure 14 below shows the Program Manager screen with the MARS icon. By clicking on the MARS icon, the user starts the application. The first screen that appears is the program initiation screen. This screen is shown in Figure 15 on page 72. It requires the user to enter a username which is

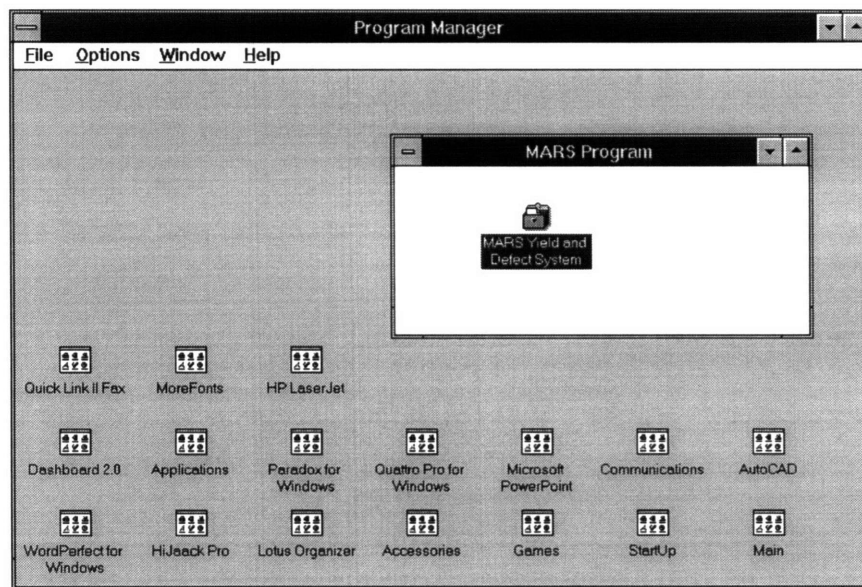


Figure 14. Program Manager Screen

used to make a spare directory on the network for the user. The program then proceeds to store a copy of the database tables in the new directory. (This is done so that the user doesn't disrupt traffic from the manufacturing floor on the network, possibly destroying the integrity of the data when he or she is querying the database

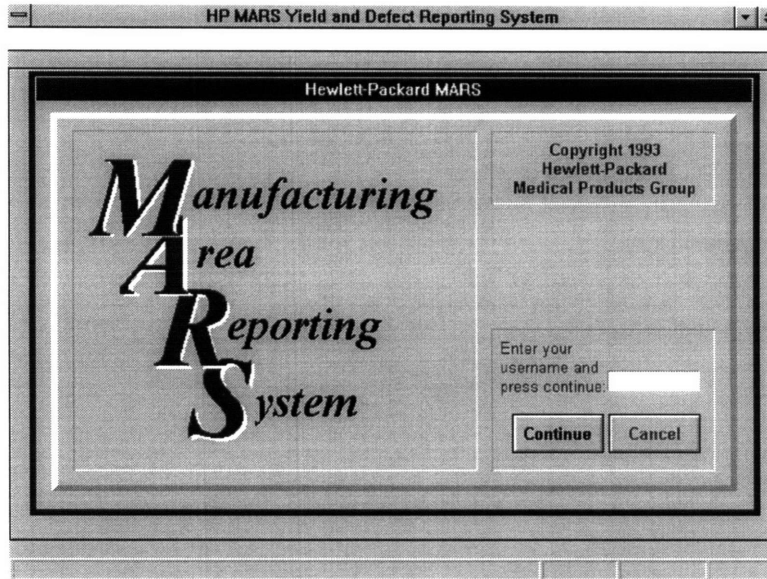


Figure 15. The opening screen of the MARS program.

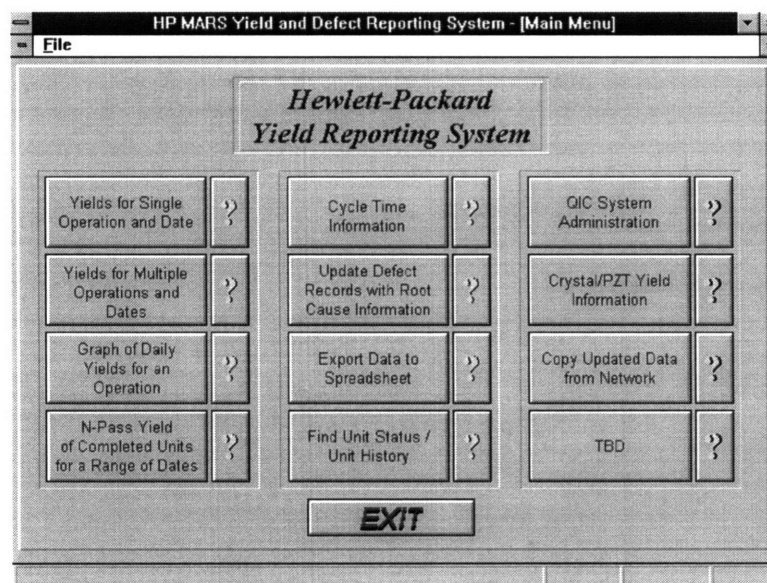


Figure 16. Main Menu for the MARS application.

and calculating yields.)

Once the program is done making a copy of the database tables, the user is given a Main Menu. Figure 16 above shows the Main Menu. From here, he or she can choose from a dozen different actions. Each of these menu choices is

summarized briefly in Appendix D on page 108.

RECOMMENDATIONS FOR FURTHER ANALYSIS

The yield reporting and defect tracking system that resulted from this investigation solved many of the problems presented in earlier sections of this report. However, there are two areas that are worthy of further study. These two areas are system ownership and database size.

System Ownership

Any system of this size needs periodic attention to ensure that it remains a successful tool. The summary statistics and yield reports obtained as output by users of the system will only be as accurate as the system's inputs. Therefore, a system administrator needs to be assigned to the MARS System to ensure that the inputs used by the application are accurate and up-to-date. Especially critical are the defect codes. As mentioned earlier, the allowable defect codes and their corresponding operation numbers are stored in a "map" that is employed in the calculation of Root Cause Yield. Unless this "map" is reliable, the Root Cause Yield Numbers produced by the system will be faulty.

Database Size

There are two concerns regarding database size that surfaced during the course of this investigation. First, as data is collected from the production floor, it is stored in database tables on a hard disk connected to a network server. This hard disk has a finite capacity. As such, periodic off-loading of the data from the network hard disk to some other media will be necessary to free hard disk capacity.

The data should be off-loaded at a frequency consistent with the needs of the Component Manufacturing Area. For example, if the people in the area regularly require reports that contain data only for the prior six months, all data collected more than six months previously could be off-loaded to save hard disk space. It is difficult to forecast what the off-loading frequency should be at this time. Once the system has been running for a few months, the frequency can more easily be determined.

Second, the speed at which the database queries are carried out is directly proportional to the database table size. To prevent the system from performing at extremely slow rates of speed, regular off-loading of the tables to some other media may be required.

III. DISCUSSION

A few weeks before arriving at Hewlett-Packard, I met with Martha Keller. Martha is a Department Manager in the Component Manufacturing Area, and she had come to MIT to discuss project possibilities with me. She outlined the current yield reporting system as discussed in the Introduction of this paper, and indicated that they needed to revise the system on the floor. We agreed that the investigation of better ways to track and report yields and defects would be a good six and one-half month project.

It seemed that there had recently been yield problems in a particular portion of the Component Manufacturing Area. It had taken the manufacturing engineers a long time to sift through all the fab files to find data that they could use. In fact, it had been two weeks before they could get a handle on the nature of the problem. This problem *definition* time could have been spent on Taguchi experiments or other problem *solution* techniques.

Within the first week after my arrival, the general sentiment in the area was clear. In the words of one employee, "HP is a computer company! There is no reason why we need a stack of paper following each unit around."

In this chapter, I will present the step-by-step progression of events that led to the conclusions drawn in the last chapter. I will begin with a brief description of the existing data infrastructure in the plant. Then, I will detail the methods of my

investigation as presented in the outline in Table V below.

**Table V
Outline for Investigation**

Overall Methodology for the Yield Reporting Investigation
1.) Information gathering <ul style="list-style-type: none">- Informal interviews- Surveys- Database programming tool selection
2.) Prototype building <ul style="list-style-type: none">- New requirements- Feedback on yield definitions- Feedback on data presentation
3.) Solution Generation and Implementation

BACKGROUND OF EXISTING DATA INFRASTRUCTURE

As I mentioned in the Chapter One, there were basically two systems for tracking yields and defects on the manufacturing floor. First, there were the logbooks at each technician's station. Second, there was an on-line system called QIC that had been installed primarily to track cycle times of different families of units.

This QIC System had been developed in the late 1980's with an outside vendor, and had proven useful in tracking yields. The system's operation was fairly simple. Each unit was scanned-in at the beginning of the manufacturing process with a hand-held bar-code reader attached to a small terminal. The bar-code reader fed the unit's unique tracking number into a Transaction Manager (a PC with special

software) that checked the validity of all entered data. Other information about each unit, such as its part number, was also entered at this stage, by way of the bar-code terminal's keyboard. If the Transaction Manger determined that all of the data was correct, the information was entered as a record into a Paradox database table that was stored on a central server.

Later, when the unit was set to leave the Component Manufacturing Area, either by being scrapped or shipped as a "good" unit to the Wire and Test Area, it would be scanned-out with a bar-code reader. The Transaction Manager PC would then compute the cycle time (throughput time) for the unit, and update the unit's information record in the database table on the server.

In the case of a scrapped unit, a four-digit defect code describing the problem would also be scanned into the QIC System. This information would be entered as a record into a second database table on the server. A list of common defect codes, and their corresponding bar-code translations, was kept at each technician's workcenter.

So, there were two Paradox database tables kept on a server that stored information about all of the units that had been built in the component area. The first table, called TUNIT_RE.DB, stored information about each unit's history, including its unique unit number, its part number, its origin, cycle time, and other data. The second table, TSCRAP.DB, stored information about all of the units that had been scrapped, including the four-digit defect code.

At the end of every month, the QIC System administrator would run a cycle-time calculation program that had been written by the same outside vendor that had

installed the QIC System. This program would automatically create a hardcopy cycle-time report, where information could be viewed in a graphical format. In addition to this, the administrator would generate a yield report using both database queries, spreadsheet macros, and software written by an outside vendor. The process of producing this report by this myriad of methods was somewhat tedious, but reasonably effective.

The yields in the report were calculated by using the information contained in TUNIT_RE.DB. By dividing the number of units that were successfully shipped in a particular time period by the total number of units that were scanned-out in that same period, a "Terminate Yield" could be calculated. A sheet containing totals of the defect codes for the scrapped units was also contained in the report. This sheet was calculated using the data in TSCRAP.DB. For a look at the yield report, see Appendix B on page 103.

The infrastructure for the QIC System was fairly strong. The database tables were well maintained by the system administrator, and people in the component area relied on the results to make decisions. Several people had become familiar with the Paradox database package through years of interfacing with the system. This fact will become important later.

INFORMATION GATHERING

I mentioned previously that the first step in my three-step investigation was to

gather information. I needed to determine the customers' needs while simultaneously investigating technologies that I thought might be able to satisfy those needs. This information gathering phase, therefore, was subdivided into three tasks. First, I interviewed key people in the manufacturing area and familiarized myself with the process of building electronic components. Second, I used the feedback and answers from these discussions to put together a survey of the technicians, engineers, and managers in the area. Third, I looked at several database programming tools and familiarized myself with one of them. In the following sections, I will detail the results of these three tasks.

Informal Interviews

I spent two weeks talking with technicians, engineers and managers in the manufacturing area. Basically, I needed to learn the formal and informal methods for transferring information on the floor. How did people do their jobs? What was involved in building a component? I began by asking four basic questions. These questions are listed in Table VI on the next page.

In addition to the standard logbook that each technician used for storing information on individual units, there were perhaps a dozen other data repositories. Some technicians took supplementary data as part of an ongoing engineering experiment. In many cases, data from completed experiments was still being taken, even though the need for the data had long past. Some technicians had begun storing

information in computer spreadsheets. This was done partially at the request of the engineering staff, and partly by the technicians' own initiative. After a time, the

Table VI Questions Regarding Manufacturing Data Collection
1.) What data is currently being kept? 2.) In what form is the data? 3.) What data is being used? 4.) What data is needed?

answers to the questions in Table VI became a little clearer. However, other questions began to surface. These new questions were best asked by means of a survey of the manufacturing personnel.

Surveys

After I started to understand the data on the manufacturing floor, I put together three surveys. Two surveys were prepared that asked questions about the needs of the customers of the new yield and defect reporting system. One survey was given to the technicians, and the other was given to the engineers. A third survey was also constructed that asked about how people in the area spent their time. I will discuss the results of these three surveys below.

Technician Yield Reporting Survey

The primary customers of any system that I was to build were the technicians

on the manufacturing line. They would be the only group of customers to interface both with the *input* and the *output* ends of the system. As such, I needed to know what they liked and disliked about their involvement in today's system for tracking and reporting yields and defects (specifically, their involvement with the QIC System), as well as what their ideas for improving the system were. I also needed to know what computer skills the technicians had. If I were to build a PC-type interface to enter and extract data, I needed to know if the technicians on the floor would be able to use it. For a copy of the actual survey, see Appendix E on page 121.

The answers to the questions were frank and informative. The top eight comments, in terms of the number of responses, were as shown in Table VII.

Table VII
Top Eight Comments Returned on
the Technician Yield Reporting Survey

- | |
|--|
| <ol style="list-style-type: none">1.) The response time of the current QIC System is too slow2.) Sometimes people do not bother to put units into the system3.) I would prefer it if we completely eliminated the logbooks and only used computers to enter the data4.) We should have PCs of our own so that we can control our own reporting5.) People don't use the QIC System properly6.) The technicians should tally their own yields daily7.) The QIC System is down a lot when needed8.) I don't get any feedback after the parts leave my area |
|--|

On average, most of the technicians also said that they felt comfortable using PCs, and many had obtained several hours of formal training.

In general, the technicians were fairly dissatisfied with the QIC System as it was currently designed. Many felt that compliance was a problem, in that people

didn't always put units into the system properly, or that people didn't bother to enter them into the system at all.

Engineer Yield Reporting Survey

The engineers were another key user of the system. As such, their needs and expectations were important, but slightly different from those of the technicians. In particular, I needed to find out what format would be most appropriate for the yield reports. Most of the engineers needed to see a lot of data for a wide variety of processes. Unless this data was presented by the system in a reasonable format, the system wouldn't be used. A copy of this survey can be found in Appendix F on page 124. (Note: this survey was also given to the managers in the area.)

I also asked them to give me a picture of what their dream system would be. In other words, I wanted to know what they would want if they could have anything. This question produced some interesting results. In reality, however, most of the answers centered around three key points. First, they felt that the system should provide information in real time. Currently, the engineers were getting information in the yield reports that was already out of date. Second, the new yield reporting system had to be flexible. The business environment was changing very fast, as were the data needs of the people who worked in the Component Manufacturing Area. Any information system built to support them would also have to be very flexible. Third, the data presented must be accurate. If there was any doubt to the accuracy of the numbers, the system would not be used, and my efforts would be wasted.

Time Survey

Any new information system costs money. Quickly getting the right information to the right people is important for a business, and money spent to that end is generally money well spent. In this case, even if the new system that I was to build was "the right thing to do" from a strategic standpoint, it would help matters if I had hard data to justify the expense of the project.

Data regarding "money saved by the system" was going to be difficult to generate. In a real sense, it is usually months or years after the implementation of an information technology project before the value of any new system is realized. Even then, the justification data is usually rather fuzzy and difficult to quantify.

To help put a value on the savings generated by investing in my project, I turned to something that was easy to quantify and relatively easy to predict: time. Every person in the Component Manufacturing Area spent time gathering data for the calculation of yields and the formation of defect Pareto Charts in some way. Technicians kept logbook data and did some calculation of first-pass yields. Engineers often spent hours pouring over old fab files looking for information. Certainly, an improved system would save time. Any time saved could either be translated directly into dollars, or into time spent doing more value-added work than mere data collection.

To help quantify the amount of time spent collecting and manipulating data in the Component Manufacturing Area, I developed a survey that asked one simple question, "How much time do you spend each week calculating yields, tracking down

unit numbers, and counting defect codes?”

The results were very interesting. Of the respondents (just over half of the component manufacturing personnel responded), each spent an average of three hours per week engaged in activity that could alternately be accomplished by an integrated yield reporting and defect tracking system. As it turns out, this would be enough to justify the expense of the system.

Programming Tools

While the technicians, engineers and managers were busy filling-out surveys, I spent some time investigating different database management packages offered by various vendors.

The database packages on the market that I could choose from were generally split into two different categories. First, there were the Unix-based packages offered by companies such as Informix, Sybase and Oracle. These packages were meant to run on workstations or enterprise servers, and were extremely powerful. A user could access database applications running these systems via an actual terminal attached to the server, or via a PC running a terminal emulation program. Most of the tools used to build database application programs for these packages were very difficult to learn. As such, the system vendors of such systems would actually do most of the programming and set-up, while the customers would provide guidance and support. In addition, most of these systems were very expensive. Engineers at HP had

previously looked at purchasing one of these systems, and had found them to be too expensive to justify.

In addition to the *enterprise-wide* solutions offered by workstation application vendors, *workgroup* solutions were available for the PC. Companies such as Borland, Microsoft, Software Associates, and Lotus sold packages that allowed small businesses, and departments of large businesses, to gather and share data. Generally, these packages were easier to learn and less expensive than their Unix-based cousins, but weren't as powerful. However, they had come a long way in terms of performance in the last several years. Due to the nature of the project, the time available to build a system, the desired cost, and the fact that PCs were readily available to run applications such as these, I chose to concentrate my investigation in the workgroup solution area instead of the enterprise-wide solution area.

One key issue is important to consider. This project was meant to be a second step toward a more powerful, all-encompassing solution to the problem of data collection and presentation in the Component Manufacturing Area. As such, the construction of a workgroup-sized solution could be seen as a valuable learning tool for the department. In the future, however, if the division is to proceed to the next step in their information strategy, and eventually reach a paperless system for data collection and retrieval (see Figure 6 on page 31), they will need to investigate enterprise-wide solutions. Hopefully, the knowledge gained from this project will help them to choose the right enterprise-wide system when the need arises.

Given that I was going to decide between available PC applications, I had four

criteria in mind in the investigation of the available packages. First, any package that I chose had to be built for the Microsoft Windows operating system. Hewlett-Packard had recently decided as a company to commit itself to having all of its PCs use Windows as the operating system. The second, third, and fourth criteria were ease of use, expandability, and strength of the programming language, respectively. I chose to evaluate Lotus Approach, Borland Paradox for Windows, and Microsoft Access. All three were Windows-based applications.

As such, all three were very easy to use. For expandability and strength of the programming language, however, Borland's Paradox for Windows was the clear winner. Borland ships an application programming language called ObjectPAL with Paradox. ObjectPAL is an object-oriented, visual programming language based on Borland's own C++ product, and it was specifically designed as a tool for developing systems such as mine. While Microsoft and Lotus included decent macro languages with their packages, neither of them could offer a full-strength application builder like ObjectPAL. Furthermore, the division already had several systems running databases in the Paradox format, including QIC.

To teach myself the ObjectPAL language, I bought a copy of the software and several books on the subject, and I consulted other database programmers at HP.

PROTOTYPE BUILDING

Once the surveys were returned, and I had chosen a database package from

which to build a solution, I began production of a prototype system. The purpose of the prototype was three-fold. First, it could uncover information about customer needs that didn't surface during the information gathering portion of the investigation. Second, it was a way to get user feedback and to check my progress toward satisfying the customers' needs. Third, it was a way for me to practice my newly-acquired programming skills.

Refining Customer Needs and Expectations

A number of key needs and expectations were further identified and refined in this section of the project. Three of the major developments involved Pareto Chart building, the definitions of yield numbers, and the formulation of a root cause assignment map.

Pareto Chart Building

The numbers of good units and scrapped units for each day were recorded in logbooks by the technicians at the particular workcenters. Scrapped units had a defect code associated with a particular type of manufacturing problem. These defect codes were entered into the QIC system when each scrapped unit was logged-out. In many cases, however, the defect codes were not entered, and the unit was therefore not officially scrapped, until a teardown analysis had been performed on the unit by another qualified technician. This process could take hours or weeks. A unit that was

taken out of the manufacturing process in mid-May for a defect might not be logged-out via the QIC system until June. Therefore, the yield numbers produced by way of QIC for May would be artificially high, while that for June would be low. Figure 17, below, shows a sample flow of the process. Some units, and their associated

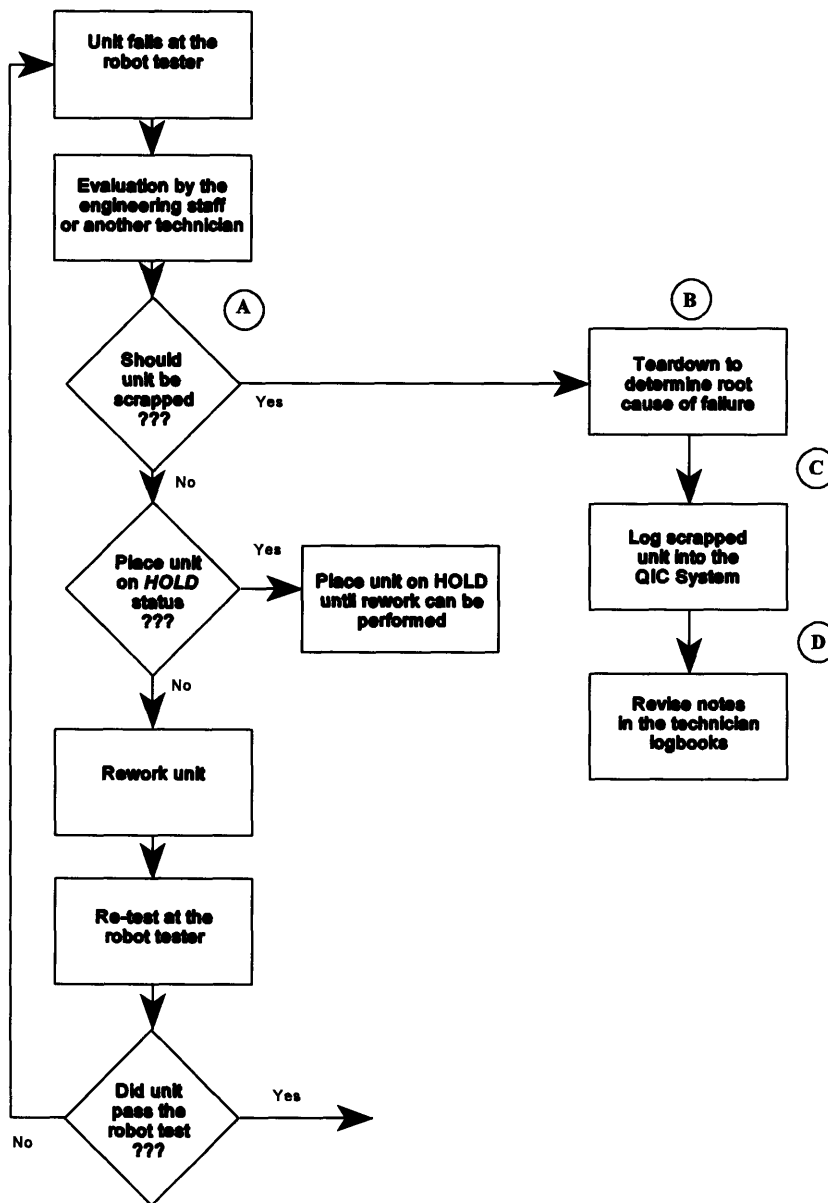


Figure 17. Sample flow of a defective unit.

paperwork, were known to be held up for days at A, B, C, or D. The root cause of the defect had to be identified before the scrap unit was logged-out via the QIC system as a terminate. There were two things that were evident about this situation that I could use in the formation of my prototype. First, it was not necessary to determine the root causes of the defects for a particular process to calculate the yield for that process. All you needed was the number of "good" units and the number of "bad" units, both of which could easily be supplied by the technician on the manufacturing floor. Second, time could be saved if the actual technicians on the floor entered information about units that were scrapped by way of the QIC system, instead of the teardown person. To do this, the technicians could determine a first-cut at a failure mode for the unit, which could be entered into the QIC system as a defect code. Later when a root cause was determined, the original defect code could be updated with the new information.

I discussed of the pros and cons of such a system for recording defects with some of the key engineers and with Martha Keller. On the plus side, the yield numbers returned under the plan would be very up-to-date. On the minus side, the actual root cause may never be entered into the system, resulting in the formation of defect Pareto Charts that were less than 100% accurate. In other words, without actual root cause analysis, Pareto Charts of the defects would be based on failure modes, not on root causes. It was decided that this situation could be tolerated if there was some sort of indication on the Pareto Charts as to the percent of the defect-codes represented that were the result of actual teardown analysis. The outcome of

this discussion was the Pareto Chart shown in Figure 18 below. Note the box in the upper middle of the screen containing the words, "percent of pareto based on actual root cause analysis." If this number is 0.0%, then the user knows that none of the units represented in the Pareto Chart have had actual root cause analysis, and that all of the information is based on failure-mode data entered by the technicians. In practice, it is expected that this percentage number would be somewhere between 25 percent and 75 percent.

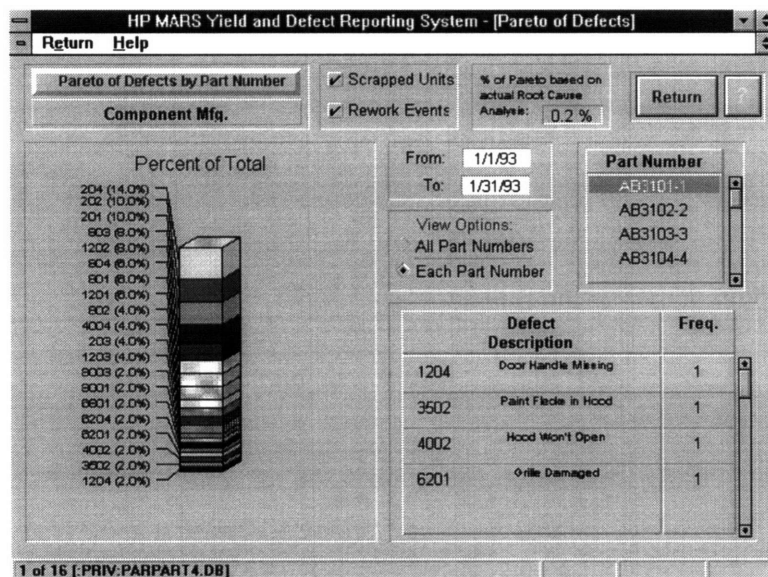


Figure 18. Pareto screen for the MARS System. Note the box indicating the percent of units receiving actual root cause analysis.

Definitions of Yield Numbers

As I began working on the yield reporting aspect of the project, I decided to build a couple of screens and show them to some key users as a way to get feedback on the format. In addition to format feedback, which I'll talk about in later sections,

another interesting result developed.

It appeared that different members had slightly different definitions of the two major yield numbers in use at the time (i.e. First-Pass Yield and N-Pass Yield).

While everyone's definitions shared a common thread, there were issues as to what constituted a scrapped unit, when units should be logged against a particular process' yields, and what to do with reworks.

I decided to interview key members of the department, as well as individuals who's definitions of the different yield numbers were very different, to better understand what should be done. After the interviews, I put together a memo outlining what I thought the shared definitions of First-Pass Yield and N-Pass Yield should be. The feedback on the report was very helpful, not only in reaching a consensus regarding the definitions of these two yield numbers, but also in illuminating their shortcomings.

It was clear that the lack of a metric describing the effect of downstream defects was leading to some of misunderstanding and misuse of the N-Pass Yield number. In addition, since the new system was supposed to be able to track reworks, it may be helpful if it also counted the number of times that a particular unit passed through a particular process before it was terminated. Based on these two problems, I decided to rough-out a definition for the Root Cause Yield metric, and for the Average Number of Reworks metric.

Armed with semi-formal definitions for First-Pass Yield, N-Pass Yield, Root Cause Yield, and Average Number of Reworks, I published a report containing the

definitions of these numbers to all of the engineers and managers in the Component Manufacturing Area. In the report, I laid out what the definitions of these four numbers would be as they related to the system that I was building. After some minor adjustments based on the feedback from this report, the final definitions, as explained on page 38, were finalized.

Root Cause Defect Map

In order for the Root Cause Yield to be calculated, the source of each possible defect encountered in the manufacture of an electronic component had to be determined. Take, for example, the illustration of the production of a model car. If the front bumper is attached at Operation 10, any defect involving the misalignment or mis-attachment of the front bumper would, therefore, be attributed to Operation 10. With the help of engineers and key technicians in the area, a spreadsheet was constructed including information on every defect for every part number and every operation on the manufacturing floor. This table, called the "Root Cause Yield Map," was translated into a database table for use by the system. It was to be updated and maintained as new types of defects were discovered, or as new part products were introduced to the manufacturing area.

Usability Feedback

In addition to the operating feedback mentioned in the last subsections, I also

solicited feedback on the user interface and overall "look-and-feel" of the prototype's various screens and reports. I wanted to make sure that the data and results were being presented in a usable, easy to understand format. Various changes were made to the screens in order to accommodate as many user suggestions as possible. In addition, I decided to provide an option to export data to a variety of spreadsheet applications. In this way, users could have even more flexibility, by providing them with the option to format the results in any way that they wished.

Programming Skill Practice

The last area of the prototype building phase involved the actual coding of the program. Since ObjectPAL is an object-oriented language, it was fairly easy to add new additions to the application in the form of "objects" without rewriting any code. This was a significant plus. In the future, as new features become important, these features can be added without changing any of the original functionality. Conversely, if any of the current functions that are included in the package become outdated, as the needs of the organization change, they can be removed without damaging the entire structure of the program. As such, I was able to add and remove whole sections of the application prototype during this phase of the project.

PROTOTYPE REFINEMENT AND SOLUTION GENERATION

In the last two subsections, I covered the first and second phases of the project in detail. As you recall, the three phases of the project were as shown in Table V. The last section to be discussed, therefore, is the solution generation and implementation phase. This phase will be discussed in the two subsections that follow.

Table V
Outline for Investigation
(repeated from page 76)

Overall Methodology for the Yield Reporting Investigation	
1.)	Information gathering <ul style="list-style-type: none">- Informal interviews- Surveys- Database programming tool selection
2.)	Prototype building <ul style="list-style-type: none">- New requirements- Feedback on yield definitions- Feedback on data presentation
3.)	Solution Generation, Implementation, and Final Project Analysis

Solution Generation

I received excellent input on building Pareto Charts, generating yield numbers, and constructing a defect map from the component manufacturing organization. In

addition, feedback on the user interface was constructive and timely. Once all of this was in place, I coded the final version of the application. In all, a prototype for every major screen had been built to demonstrate the functionality of the different parts of the system. All that was left to do was to tie these together, and to make sure that the resulting application interfaced well with the network.

The network on which this system was supposed to exchange data was a local area network running Microsoft's LANManager software. The most difficult issue with operating and managing a database across a LAN is data integrity. With several users potentially accessing the same data at once through the MARS system at the same time, the program had to ensure that the data wasn't compromised. To solve this problem, I consulted with other LAN and database managers at the division, as well as people who had originally installed the QIC System.

The solution was to make a copy of the data for each user of the MARS program at the moment that the program was started on the user's PC. This copy would be stored in a separate directory, created by the application program, on the network. The original data was left intact in its own directory, thereby freeing it for exclusive use by the Transaction Manager PC. If, during a session on the MARS System, the user wished to update his or her private directory with new data from the "live" database file, a button on the main menu would facilitate the request. When the user was finished calculating the necessary yields (or whatever he or she logged-on to do), the program would erase his or her copy of the data and delete the separate directory.

The final application program was made up of thirty-two separate screens. For an explanation of some of the key functionality, see Appendix D on page 108. For an overview of the screens, see Appendix G on page 126.

Implementation

Implementing a new information system is a non-trivial exercise, especially when it involves a re-engineering of the whole process that the information system was built to support. Thorough training of the users is not enough to guarantee a successful implementation. Rather, a true change in the culture is often necessary. In this instance, the technicians on the floor were to be propelled into a whole new level of involvement with the collection of data and the generation of yield reports.

Because this new system was a radical change in the day-to-day activity of an technician, and training was going to be extensive, a good implementation plan had to be developed. I worked with Yves Champagne, a new engineer in the Component Manufacturing Area, and Dexter Daniel, a developer on the system, to construct a plan for installation. As a team, we solicited input from the technicians, engineers, and managers. The main point of including this information here is not to talk about the details of the plan, but to point out that the actual formation of a plan is an important step.

Final Project Analysis

Table VIII, below, gives a summary of the differences between the system developed under this project and the old system for generating yield and defect reports.

**Table VIII
Comparison of Old QIC System
to MARS Project System**

Old QIC System	MARS Project System
<p>Yield Reports</p> <ul style="list-style-type: none"> - No automatic yield reports are generated, only cycle time reports. - Yield numbers can be generated manually from QIC, but only for an entire manufacturing area – not for individual operations. For individual operations, yield numbers are calculated by manually gathering data from technician logbooks and entering them into a spreadsheet program. 	<p>Yield Reports</p> <ul style="list-style-type: none"> - Yield reports available at the touch of a button for any combination of dates, part numbers, and operations.
<p>Yield Types Considered</p> <ul style="list-style-type: none"> - The current yield reports generated are for Terminate Yields. Basically, the only information available on QIC is for units that have left the manufacturing area –either by being scrapped or shipped. 	<p>Yield Types Considered</p> <ul style="list-style-type: none"> - Information available for First-Pass Yield, N-Pass Yield, Root Cause Yield, and Average Number of Reworks for any combination of dates, part numbers, and operations. - The ability to generate Terminate Yields is also included.

Old QIC System	MARS Project System
<p>Time-based Graphs of Yield Numbers</p> <ul style="list-style-type: none"> - None are currently generated automatically. 	<p>Time-based Graphs of Yield Numbers</p> <ul style="list-style-type: none"> - Graphs of daily yields for a period of up to two months can be generated at the touch of a button. - Efforts of quality improvement teams can be easily measured using trend graphs of yields for a particular part number and/or operation.
<p>Rework Tracking</p> <ul style="list-style-type: none"> - Rework information in the Component Manufacturing Area is recorded on individual bug sheets in the fab file, but is not kept in a central location. 	<p>Rework Tracking</p> <ul style="list-style-type: none"> - Central location for tracking rework and rework yields is provided. - The system keeps track of rework events and rework cycles, allowing engineers to keep track of the amount of rework being performed.
<p>Pareto Charts of Defects</p> <ul style="list-style-type: none"> - Counts of scrap defects are generated through a complex set of spreadsheet macros from the QIC data. - No Pareto Charts of rework events are available. 	<p>Pareto Charts of Defects</p> <ul style="list-style-type: none"> - Pareto of scrap defects can be easily generated at the touch of a button for any combination of part numbers, build dates, operations, etc. - Pareto of rework events can be generated just as for scrap events.
<p>Root Cause Analysis</p> <ul style="list-style-type: none"> - Results of individual teardowns and root cause analyses are primarily kept in files at each engineer's desk. 	<p>Root Cause Analysis</p> <ul style="list-style-type: none"> - Results of teardowns and root cause analysis can be stored in a central location, allowing everyone to share the information more easily.

Old QIC System	MARS Project System
<p>Spreadsheet Compatibility</p> <ul style="list-style-type: none"> - Data from the floor must be manually entered into a spreadsheet program. - QIC provides some ability to export information to a Lotus 1-2-3. However, the exporting is nothing more than a data dump, with little ability to filter the data. 	<p>Spreadsheet Compatibility</p> <ul style="list-style-type: none"> - Data from floor can be automatically downloaded to a spreadsheet program for further manipulation. Filters are available so that only the necessary information is exported. - Data can be exported to Lotus 1-2-3, Borland Quattro Pro, and Microsoft Excel. Data can also be ported to Informix, Sybase, Oracle, and other standard database formats.
<p>Unit Tracking</p> <ul style="list-style-type: none"> - Finding a particular unit on the floor cannot be achieved via QIC. Instead, logbooks at each operation are manually checked to see if the unit has passed through the operation. 	<p>Unit Tracking</p> <ul style="list-style-type: none"> - The status of an incomplete unit on the floor can be easily examined. Unit location is tracked to the operation where it was last seen.
<p>Unit History</p> <ul style="list-style-type: none"> - Historical information on completed units is hard to gather, since the data is stored in many different locations. 	<p>Unit History</p> <ul style="list-style-type: none"> - A comprehensive history of each completed unit can be obtained at the touch of a button, including each operation the unit passed through, rework cycles, hold events, etc.

As the reader can see from the table, the MARS System incorporates several measurable improvements over the old QIC System. Currently, the new MARS System is in place at Hewlett-Packard.

BIBLIOGRAPHY

"Concept Engineering", Cambridge, MA: Center for Quality Management, 1992.

Hammer, Michael, "Reengineering Work: Don't Automate, Obliterate", *Harvard Business Review*, July-August, 1990, page 104.

Simpson, Alan, *Mastering Paradox for Windows*. Alameda, CA: Sybex, 1993.

Tinney, Diane, *Paradox for Windows Programming by Example*. Carmel, IN: Que, 1993.

APPENDICES

APPENDIX A

Illustration of the Manual Yield Report*

TOY CAR MANUFACTURING MAY 1993												
	Car Model A			Car Model B			Car Model C			Car Model D		
	Good	Bad	%	Good	Bad	%	Good	Bad	%	Good	Bad	%
Attach Frt. Bumper	4	1	80	10	1	91	11	1	92	12	2	86
Attach Rear Bumper	5	2	71	15	1	94	12	2	86	14	3	82
Attach Doors	6	1	86	18	2	90	13	1	93	16	2	89
Attach Hood	7	2	78	13	2	87	15	1	94	13	1	93
Composite Yield	38%			66%			68%			58%		
Summary												
Total throughput	48											
Weighted Average Yield	61%											
<hr/>												
	Car Model E			Car Model F			Car Model G			Car Model H		
	Good	Bad	%	Good	Bad	%	Good	Bad	%	Good	Bad	%
Attach Frt. Bumper	5	1	83	10	1	91	7	1	88	2	2	50
Attach Rear Bumper	6	2	75	14	3	82	5	3	63	3	3	50
Attach Doors	4	1	80	15	2	88	4	1	80	2	2	50
Attach Hood	6	2	75	12	1	92	6	1	86	5	1	83
Composite Yield	38%			61%			38%			10%		
Summary												
Total throughput	89											
Weighted Average Yield	43%											
<hr/>												
	Car Model I			Car Model J			Car Model K			Car Model L		
	Good	Bad	%	Good	Bad	%	Good	Bad	%	Good	Bad	%
Attach Frt. Bumper	14	1	93	11	3	79	21	2	91	12	1	92
Attach Rear Bumper	15	1	94	12	1	92	22	2	92	14	0	100
Attach Doors	16	1	94	10	2	83	23	6	79	16	0	100
Attach Hood	17	3	85	10	1	91	25	4	86	13	0	100
Composite Yield	70%			55%			57%			92%		
Summary												
Total throughput	85											
Weighted Average Yield	67%											
<hr/>												
Overall Sensor Summary												
Total Throughput	142											
Weighted Average Yield	54%											

**actual figures have been disguised*

APPENDIX B

Illustration of Semi-Auto Report*

QIC SYSTEM - Units Scrapped 05/01/93 - 05/31/93 inclusive

Model	# Scrapped	Total	Yield %
Car Model A	6	13	54%
Car Model B	6	19	68%
Car Model C	5	20	75%
Car Model D	8	21	62%
Car Model E	6	12	50%
Car Model F	7	19	63%
Car Model G	6	12	50%
Car Model H	8	13	62%
Car Model I	6	23	74%
Car Model J	7	17	59%
Car Model K	14	39	64%
Car Model L	1	14	93%

KEY

1010	Front Bumper Not Aligned
1020	Front Bumper Fell Off
1030	Wrong Bumper on the Front
2010	Rear Bumper Not Aligned
2020	Rear Bumper Fell Off
2030	Wrong Bumper on Rear
3010	Dent in Door
3020	Door Misaligned
4010	Hood Not Aligned
4020	Hood Fell Off
4030	Dents in Hood

	Attach	Front	Bump.	Attach	Rear	Bump.	Attach	Doors	Attach	Hood	
Model	1010	1020	1030	2010	2020	2030	3010	3020	4010	4020	4030
Car Model A	1			1		1	1		2		
Car Model B		1			1		1	1		2	
Car Model C		1				2		1	1		
Car Model D	1		1	2	1		2				1
Car Model E	1				2		1		2		
Car Model F		1		1	1	1		2		1	
Car Model G			1								
Car Model H		2			2	1		1	1		
Car Model I	1			1			1		1	1	1
Car Model J	1	2			1			2	1		
Car Model K			2		2		3	3		2	2
Car Model L	1										

**figures have been disguised*

APPENDIX C

Illustration of the Application of Different Yield Numbers

For the example, I will take eight parts, each with a different history, that were processed at Operation 10 on January 7. It is now February 1, and we want to know the values of the aforementioned four metrics for January 7. The four parts entered the operation on the morning of January 7 as described in Table C1.

Table C1
Entering Status of Four Sample Parts

Unit Number	Entering Status
A1001	Fresh part. Just arrived from Operation 5 last afternoon.
A1002	Fresh part. Just arrived from Operation 5 last afternoon.
A1003	Fresh part. Just arrived from Operation 5 last afternoon.
A1004	Fresh part. Just arrived from Operation 5 last afternoon.
A1005	Fresh part. Just arrived from Operation 5 last afternoon.
A1006	Fresh part. Just arrived from Operation 5 last afternoon.
A1007	Fresh part. Just arrived from Operation 5 last afternoon.
A1008	Rework. This part originally passed through the operation on January 4. Now, Operation 70 has sent it back to be reworked.

Today, February 1, we can look back and find out what actually happened to the eight parts. Table C2 gives the results.

APPENDIX C

Illustration of the Application of Different Yield Numbers *(continued...)*

Table C2
Exit Status of Eight Sample Parts

Unit Number	Status as of the end of the day January 7	Status as of February 1
A1001	Finished Operation 10 successfully on the first pass. Shipped to next operation.	Scrapped at Operation 80 due to an error made by an operator at Operation 80.
A1002	Finished Operation 10 successfully on the first pass. Shipped to next operation.	Scrapped at Operation 80 for a defect related to Operation 10.
A1003	First pass - needed rework Second pass - needed rework Third pass - finished successfully and shipped to next operation.	Eventually shipped to Wire and Test.
A1004	First pass - needed rework Second pass - needed rework Third pass - scrapped for a defect related to Operation 10.	Same. Still scrapped.
A1005	Finished Operation 10 successfully on the first pass. Shipped to next operation.	Eventually shipped to Wire and Test.
A1006	Work started. Unit placed on <i>HOLD</i> to wait for an engineering evaluation. Still on <i>HOLD</i> .	Taken off of <i>HOLD</i> , finished successfully in the first pass on January 11. Eventually shipped to Wire and Test.
A1007	Scrapped due to a defect related to Operation 5.	Same. Still scrapped.
A1006	Reworked Successfully. Shipped to next operation.	Eventually shipped to Wire and Test.

From this, we can calculate a First Pass Yield, N-Pass Yield, Root Cause Yield, and Average Number of Passes for Operation 10. The four metrics calculate as shown in Table C3 on the next page.

APPENDIX C

Illustration of the Application of Different Yield Numbers (continued...)

**Table C3
Metrics for January 7**

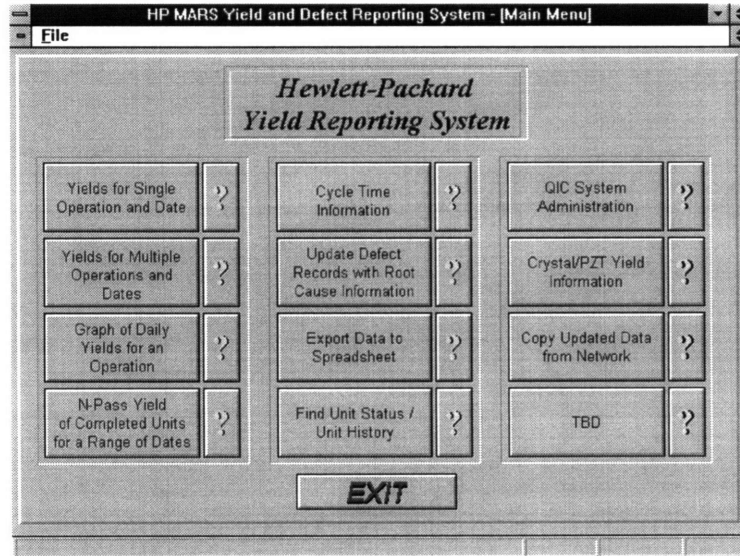
Metric	Value	Reason
First Pass Yield	4/7 = 57.1%	<p>There are only seven units in the denominator because there were only seven fresh starts at Operation 10 on January 7. (Unit A1008 was a fresh start at the operation on January 4, and had come back to the operation for rework.)</p> <p>There were only four units that were successfully completed on the first pass. Even though unit A1006 went on <i>HOLD</i> until the following day, it only took one "pass" to complete the operation. (The <i>HOLD</i> step is not counted as a separate pass.)</p>
N-Pass Yield	5/7 = 85.7%	<p>There are only seven units in the denominator because there were only seven fresh starts at Operation 10 on January 7. (Unit A1008 was a fresh start at the operation on January 4, and had come back to the operation for rework.)</p> <p>Unit A1004 and unit A1007 were not able to be shipped to the next operation because they were scrapped. Even though unit A1007 was scrapped due to a defect related to Operation 5, it is not charged back to Operation 5. This leaves five out of the seven parts to be labelled as "good."</p>

Metric	Value	Reason
Root Cause Yield	5/7 = 85.7%	<p>There are only seven units in the denominator because there were only seven fresh starts at Operation 10 on January 7. (Unit A1008 was a fresh start at the operation on January 4, and had come back to the operation for rework.)</p> <p>Unit A1002 was scrapped at Operation 80 for a defect related to Operation 10. Unit A1004 was scrapped at Operation 10 for a defect related to Operation 10. The other five parts count as "good" for the purposes of figuring Root Cause Yield.</p> <p>Unit A1007 was indeed scrapped, but it was scrapped for a defect related to Operation 5.</p>
Average Number of Passes	11/7 = 1.6 passes	<p>A1001 - one pass A1002 - one pass A1003 - three passes A1004 - three passes A1005 - one pass A1006 - one pass A1007 - one pass A1008 - zero passes. A1008 doesn't count because it was a fresh start at the operation on January 4, and had come back to the operation for rework. TOTAL = 11 passes</p> <p>There are only seven units in the denominator because there were only seven fresh starts at Operation 10 on January 7. (Unit A1008 was a fresh start at the operation on January 4, and had come back to the operation for rework.)</p>

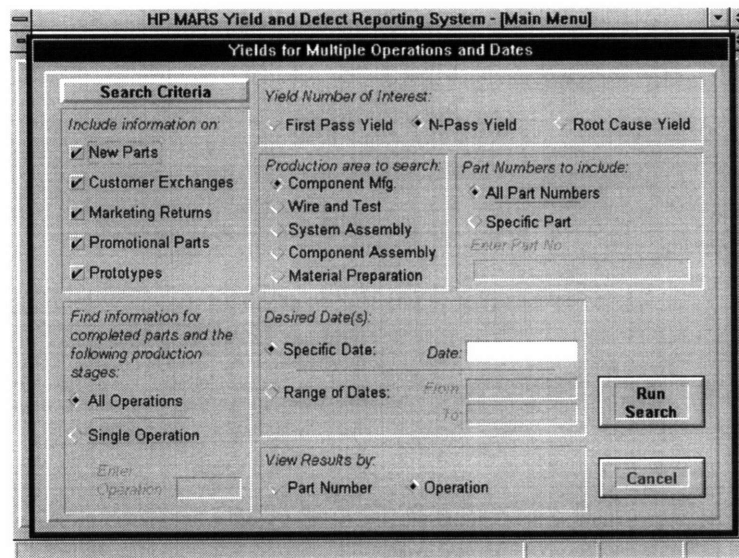
APPENDIX D

Sample usage the of the MARS application

Yield for Multiple Operations and Dates



The main menu screen above is what appears after the password screen. There are 12 choices available to the user.



When the user selects the choice "Yields for Multiple Operations and Dates," this screen appears, allowing the user to enter choices for the database search. He or she can check boxes that tell the program whether to include new parts or just prototypes, etc. For this, we will ask for information on every type of part. In addition, the user can choose between N-Pass, First Pass, and Root Cause Yield, and can choose between including all operations in the calculations, or merely one.

APPENDIX D

Sample usage the of the MARS application

Yield for Multiple Operations and Dates (continued...)

The screenshot shows the HP MARS Yield and Defect Reporting System interface. The title bar reads "HP MARS Yield and Defect Reporting System - [N-Pass Search Results by Operatio]". The menu bar includes "Search", "Return", "Report", "Pareto", and "Help".

The main window is titled "N-Pass Yields by Operation". It contains several sections:

- Component Mfg.:** A text field for component manufacturing information.
- Requested Dates:** Fields for "From:" (1/1/93) and "To:" (1/31/93).
- Product Types:** A list of checkboxes for "New", "Marketing Return", "Customer Exchange", "Promotion", and "Prototype", all of which are checked.
- Buttons:** "New Search", "Main Menu", "Defect Pareto", "Print Report", and "HELP ?".
- Operation List:** A vertical list of operations: 2, 8, 9, 10, 12. Operation 2 is currently selected.
- Data Table:** A table with columns: Part Number, Number of Starts, Good Units, Terminated Units, N-Pass Yield, and Average Number of Passes. The data is as follows:

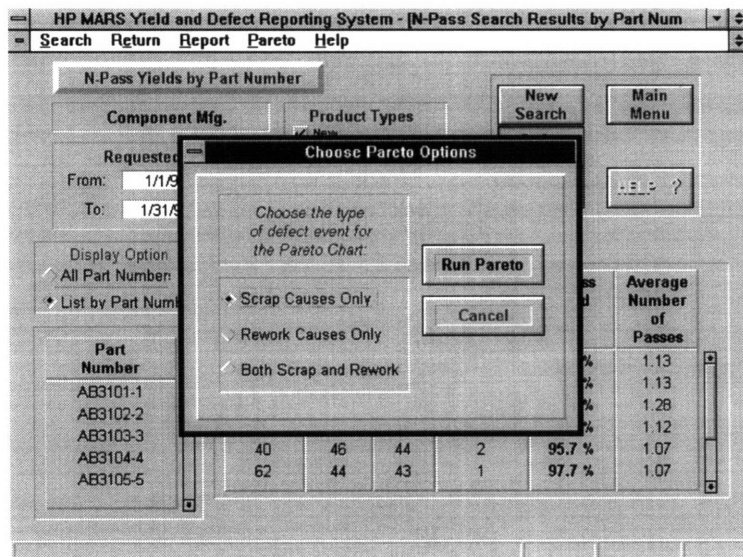
Part Number	Number of Starts	Good Units	Terminated Units	N-Pass Yield	Average Number of Passes
AB3101-1	62	55	7	88.7 %	1.13
AB3103-3	62	60	2	96.8 %	1.08
AB3104-4	60	55	5	91.7 %	1.10
AB3105-5	61	57	4	93.4 %	1.08
AB3106-6	69	61	8	88.4 %	1.06
- Summary:** A row at the bottom of the table showing totals: 314 (Number of Starts), 288 (Good Units), 26 (Terminated Units), 91.7 % (N-Pass Yield), and 1.09 (Average Number of Passes).

The status bar at the bottom left indicates "1 of 22 [PRIV:OPPN2.DB]".

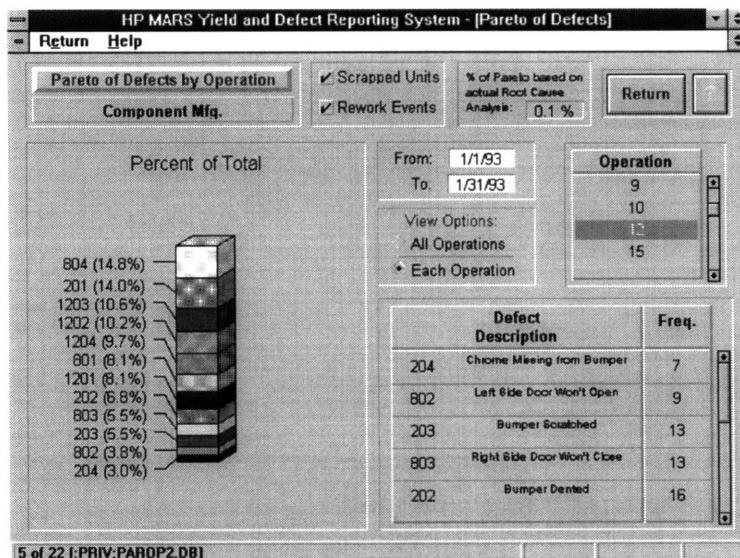
After the computer searches the database, the following screen appears, giving information about each part number for each operation. The user can use the slide bars on the tables to view the results for each operation or part number. A summary of the results for all part numbers corresponding to each operation is given at the bottom.

APPENDIX D Sample usage the of the MARS application

Pareto Chart for Single Operation and Date



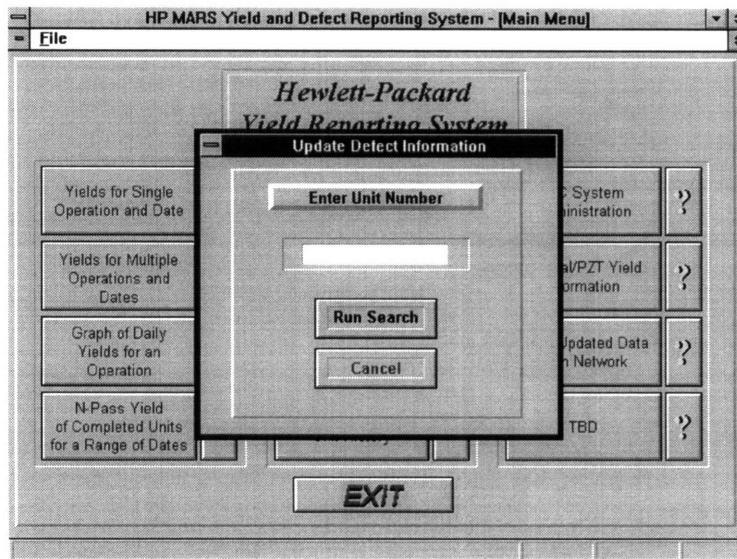
When the user selects the choice "Defect Pareto," this criteria selection appears. The user can then select whether to include scrapped units only, rework event, or both.



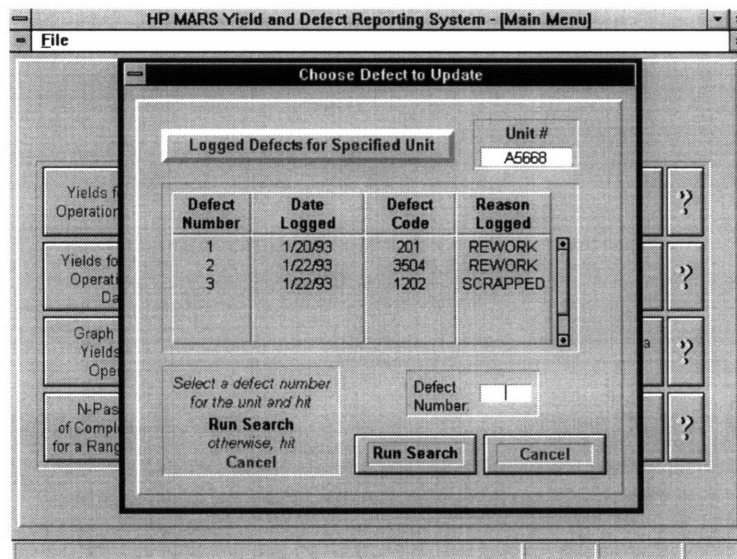
The computer then produces a screen with a Pareto Chart like the one shown above. The chart includes a percentage breakdown of each defect code. Some of the defect codes in the chart are revised codes that have been updated based on root cause analysis. The rest are merely the failure mode codes that were entered by the line technicians at the time that the defect was discovered. To help the user, a box at the top of the screen gives the percentage of the defects represented in the chart that are based on actual root cause analysis.

APPENDIX D Sample Usage the of the MARS Application

Updating Defect Information



When the user selects the choice "Update Defect Records with Root Cause Information," the input screen above appears. The user then enters the unit number of the unit that he or she wishes to update.



When the computer finishes searching the database for the unit number, this screen is presented. If there is more than one defect for the unit number in question, then the user merely selects the one that he or she wishes to get information about.

APPENDIX D Sample Usage the of the MARS Application

Updating Defect Information (continued...)

HP MARS Yield and Defect Reporting System - [Update the Defect Record]

Search Return Report Help

Update Defect Information with Root Cause Data

Hit the **UPDATE** key to edit or update the cause code and detailed defect Information.

New Search Main Menu
Print Report HELP?

Unit Number: A5668 Defect No.: 2 Defect Logged by: Ed Lee
Status: REWORK Original Log Date: 1/22/93 Part Number: AB3106-6
Defect Code Originally Logged: 3504 Operation where Defect was Logged: 40

UPDATE Defect Cause Code: 3504
Defect Reference Location: Side 1
Part Number of Defective Component: 1234-1234
Serial/Lot Number of Defective Component: 123ABC
Comments: Messed Up

Box is checked if Root Cause information has been previously updated

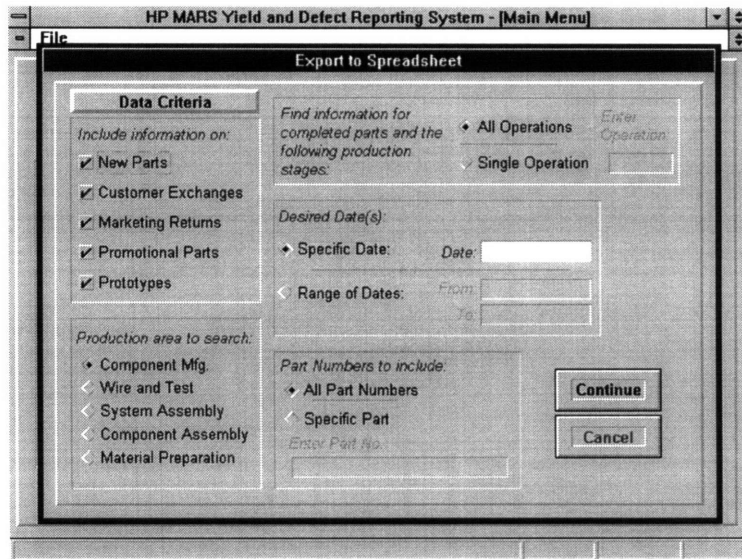
1 of 1 [PRIV:DFCTCHG8.D8]

Next, the user is presented with this screen. By pressing the "Update" button, he or she can change the information in the lower box. The defaults shown were entered by the technician on the line when the unit was scrapped.

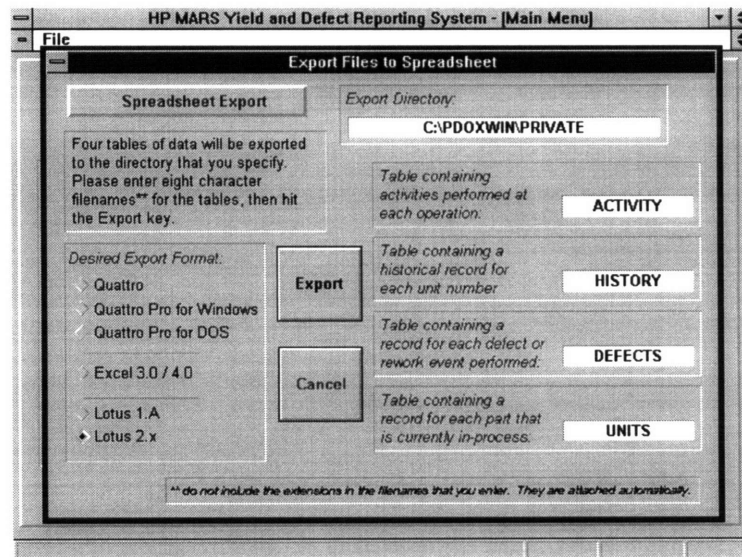
If the record is successfully changed, the box in the lower right-hand corner of the screen receives a check mark, signifying that the record has indeed been updated.

APPENDIX D Sample Usage the of the MARS Application

Exporting Database Information



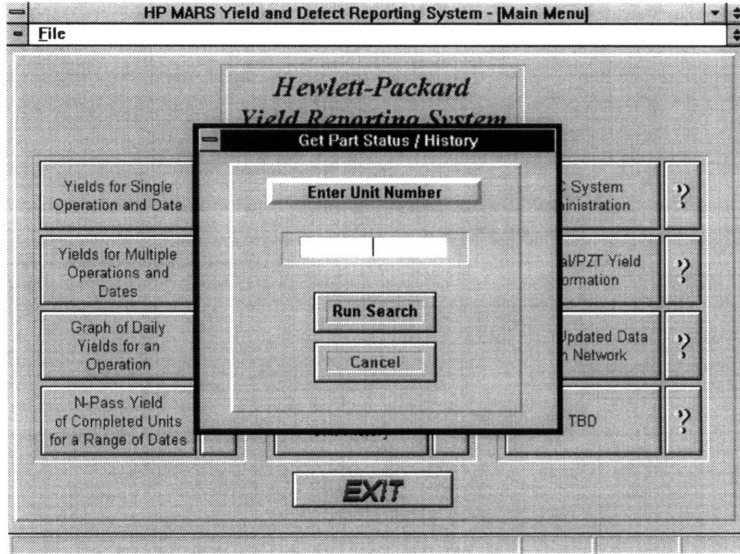
When the user selects the choice "Export Data to Spreadsheet," the screen above appears. The user then can enter various criteria for the units about which he or she wants to export information.



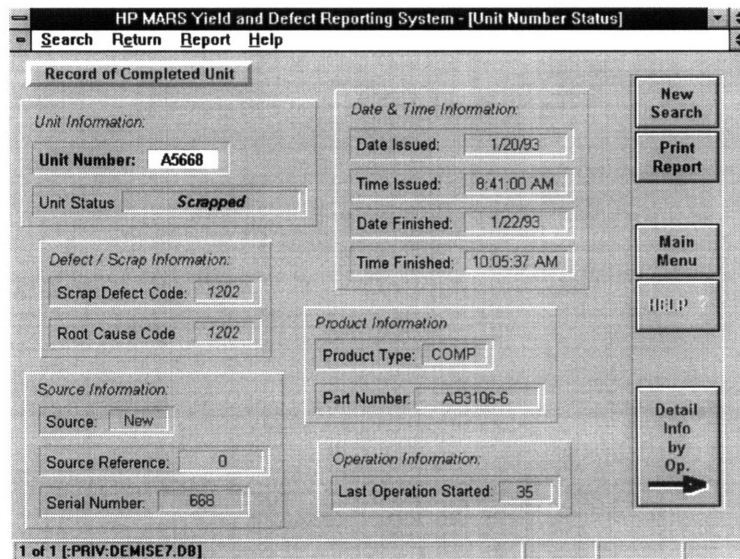
After the computer is finished searching the database, the following screen appears. Fields to enter the filenames corresponding to the different types of information are presented with default values. The user can select from six different export formats.

APPENDIX D Sample Usage the of the MARS Application

Unit Status / History



If the user wants to get information about a particular unit, he or she can hit the button for "Find Unit Status / Unit History." The screen above appears, allowing the user to enter the unit number.



If the unit has gone through the entire operation, the screen above appears.

APPENDIX D Sample Usage the of the MARS Application

Unit / Status History (continued...)

HP MARS Yield and Defect Reporting System - [Detailed Status Information]

Return Help

Detailed Operation

Unit Number: **A5668**

Return to Last Screen
Help ?

Successfully Completed Operations	Operation	Start Date	Start Time	Finish Date	Finish Time
	2	1/20/93	12:00:00 PM	1/20/93	1:01:52 PM

Unsuccessful Operations Resulting in Rework	Operation	Date Sent on Rework	Time Sent on Rework	Cause Code	Had Root Cause Analysis
	8	1/20/93	11:00:57 AM	201	[]

Operations that were Reworked	Operation	Date Rework Started	Start Time	Date Rework Finished	Finish Time
	2	1/20/93	4:07:30 PM	1/21/93	9:16:52 AM

Hold Events	Operation	Hold Start Date	Hold Start Time	Hold Finish Date	Hold Finish Time	Hold Code
	40	1/22/93	12:57:50 AM	1/22/93	12:00:24 AM	4000

Instructions:
Use the arrow buttons to scroll through the production history of the specified Unit.

1 of 4 [-PRIV:GOODS7.DB]

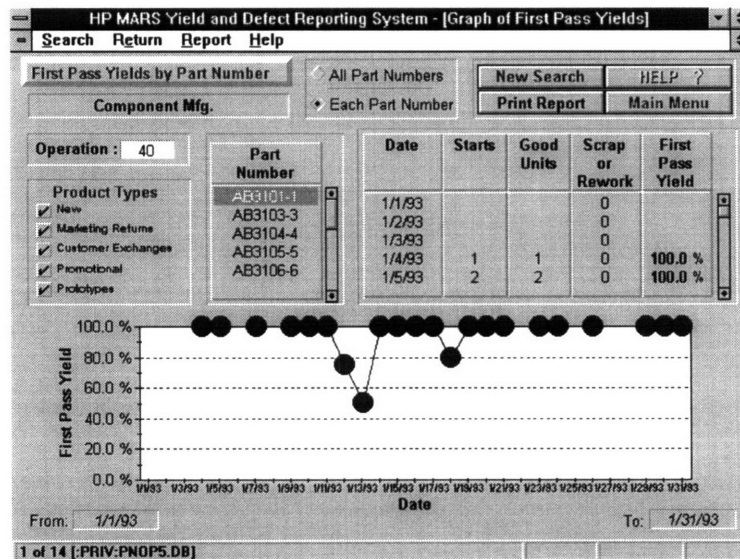
The button in the lower right-hand corner of the preceding screen brings up this screen. Information for each operation that the unit has been through under various circumstances is given. The up and down arrow keys by each table can be used to scroll through the operations.

APPENDIX D

Sample usage the of the MARS application

Graph of Yields for a Range of Dates

When the user selects the choice "Graph of Daily Yields for an Operation" from the main menu, the screen above appears. By checking the appropriate boxes and filling in the fields, he or she can define the units that will be included in the calculation.

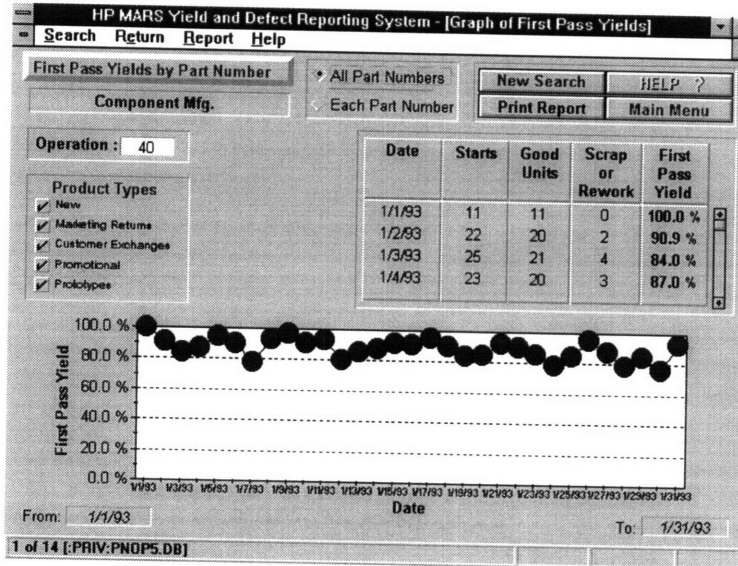


After the computer searches the database and calculates the yields, the screen above appears. The slide bars on each table allow the user to select the part numbers of interest.

APPENDIX D

Sample usage the of the MARS application

Graph of Yields for a Range of Dates (continued...)

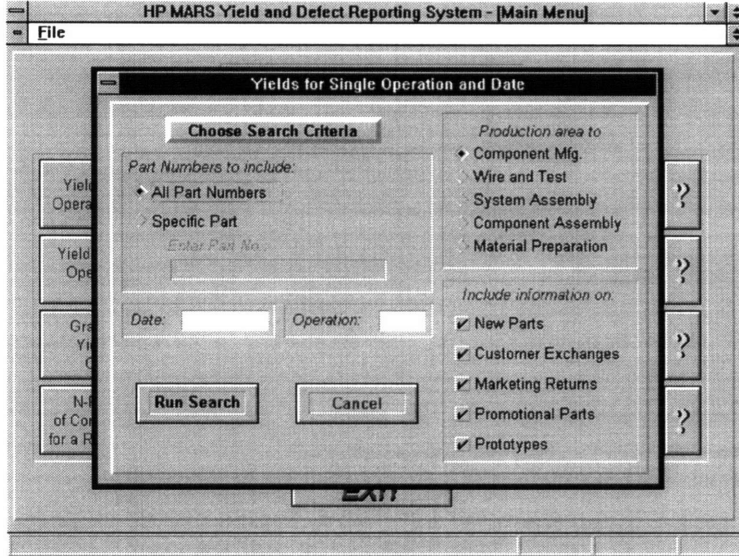


At the top center of the screen, the user can select whether to display a graph of all part numbers, or a graph for each individual part number. The graph shown here is for all part numbers.

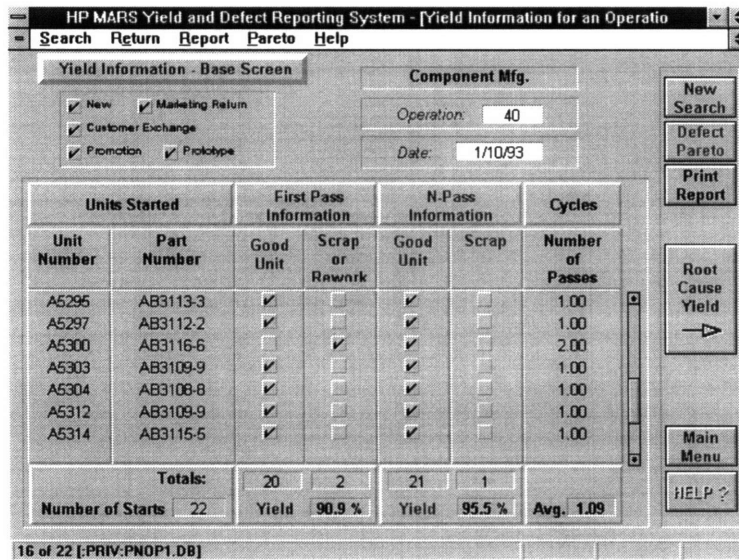
APPENDIX D

Sample usage the of the MARS application

Yield for a Single Operation and Date



When the user selects the choice "Yields for Single Operation and Date," the input screen above appears. By making the desired selections, the program knows which units to include in the calculations.



This screen appears next to give the user info on the units that passed through the chosen operation on the particular day in question.

The scroll bars at the right of the table can be used to move through the list of units.

This screen most closely replaces the technician logbook sheets in use before the new system was put into place.

APPENDIX D

Sample usage the of the MARS application

Yield for a Single Operation and Date (continued...)

The screenshot displays the 'HP MARS Yield and Defect Reporting System - [Root Cause Yield Information]' window. It features a 'Return' and 'Help' menu at the top. The main content is divided into two sections: 'Yield Summary' and 'Yield Detail'.

Yield Summary

Number of Starts	Good Parts	Parts Scrapped for Related Defects	Root Cause Yield
22	21	1	95.5 %

Percent of Scraps with Root Cause Analysis: 0.0 %

Yield Detail

Units that started operation 40 on 1/10/93, went on to be scrapped for Cause Codes associated with that operation as shown in the table below.

Unit Number	Part Number	Operation Where Scrapped	Defect Root Cause	Date Unit was Scrapped
A5304	AB310B-B	50	4003	1/10/93

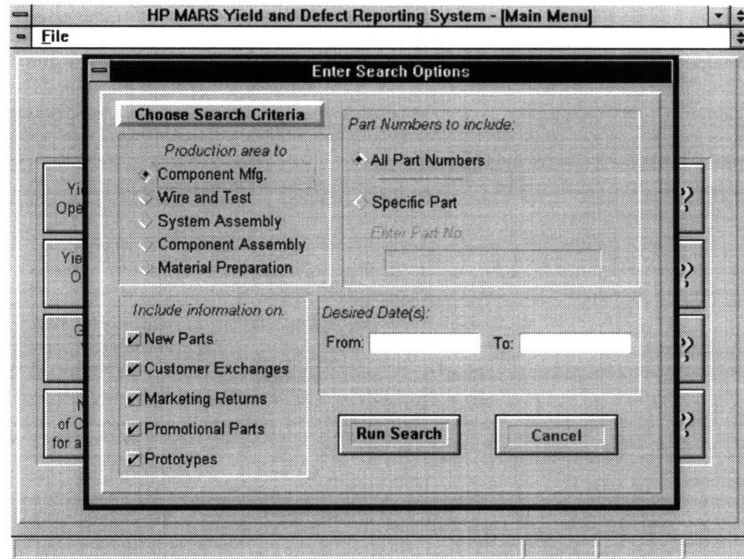
1 of 1 [-PRIV:CAUSE1.D8]

By pressing the "Root Cause Yield" button, the screen above is displayed. The user (normally a line technician) can get a feel for the success of the particular units at other operations.

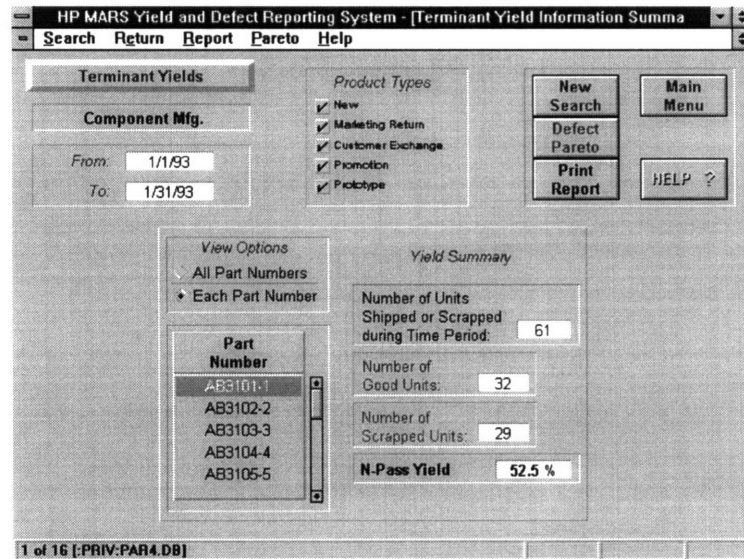
APPENDIX D

Sample usage the of the MARS application

N-Pass Yields for a Range of Dates (Terminant Yields)



When the user selects “N-Pass Yield of Completed Units for a Range of Dates” from the main menu, the screen above is displayed. Using this input screen, the user can select the criteria that he or she wishes for the search calculations.



After the user searches the database and performs the calculations, this screen appears. The user can then select whether to view the data for each part number, or for all part numbers.

APPENDIX E

Yield Reporting System Survey Given to the Technicians

TO: Component Assembly People
FROM: Ed Lee
DATE: June 28, 1993
SUB: **Yield reporting system for manufacturing area**

Hello

As you already know, I will be here for the next few months working on an enhanced yield-reporting and defect-tracking method for the manufacturing area. I've spent the last couple of weeks informally interviewing people and learning some of the ins and outs of the process. Now, its time to get down to business and start to get this thing done.

Initially, I have the idea to put together some sort of computer based tool to access data from the floor. On your end, data on operations and yields would be entered via QIC or some other screen menu. Summaries of daily, weekly, or monthly yield and defect information would then be made available to you so that you can better track the quality of your process. Anything is possible for this system. This is where I need your help.

I want to make sure that anything I do *will actually be useful*. Therefore, I need to know what kinds of data that you folks actually need and use. Please take a few minutes to fill out this survey. The questions here are rather open-ended, and there will probably be a wide range of answers (or -- maybe not). I can't promise to please everyone here, but knowing some of this information will put me on the right track.

The Survey

This survey should only take a couple of minutes. I will keep all answers to myself, so feel free to be frank and honest. Also, don't be afraid to get carried away and wordy, the more information I have, the better. I have handed out a similar survey to the engineers and supervisors, since they will also have to live with the system.

APPENDIX E

Yield Reporting System Survey Given to the Technicians

Name: _____

Process Responsibility: _____

- 1.) *How do you currently keep track of yields for your specific process?*

- 2.) *Do you provide or receive summaries or monthly totals on yields for your process. If you provide a summary of some kind, how is it done. (For example, "I add up the totals for the month and hand the results to Janice.")*

- 3.) *What suggestions do you have to improve the layout or format of the logbooks so that they would be easier to use?*

- 4.) *Often, products with defects triggered by one process have these defects found by other processes down the line. How is information on these "downstream defects" fed back to you for your processes? How might you like it done differently?*

- 5.) *Do you use QIC? _____ YES _____ NO*

If yes, please answer questions a.), b.), and c.). If no, go to question 6.)

- a.) **What are your three biggest complaints about the system?**
- b.) **What new defect or hold codes should be added to the system?**
- c.) **If you had a choice for a method to input data, which one would work best for you?**

_____ Bar code readers

_____ Computer screen and keyboard

_____ Option to use either bar code reader or computer

APPENDIX E

Yield Reporting System Survey Given to the Technicians *(continued...)*

6.) *In order for me to get a feel for everybody's computer ability, please check the answer that best describes your familiarity with personal computers.*

- | | | |
|---------------------|--------------------------|---|
| | <input type="checkbox"/> | I can barely find the ON switch. |
| Beginner | <input type="checkbox"/> | I've used personal computers, but I can't really remember much about them. |
| | <input type="checkbox"/> | I use PCs, but I'm not familiar with software programs like Lotus or WordPerfect. |
| Intermediate | <input type="checkbox"/> | I am a regular user of PCs and most of the common software programs. |
| | <input type="checkbox"/> | I can use all of the popular software programs like the back of my hand, and I also write some of my own software applications. |
| Advanced | <input type="checkbox"/> | I have a PhD in Computer Science. |

7.) *Have you ever used Microsoft Windows on a PC before?*

- Never used it.*
- Learned it, but don't use it.*
- Use it sometimes.*
- Use it every day.*

APPENDIX F

Yield Reporting System Survey and Given to Engineers and Managers

TO: Manufacturing Engineers
FROM: Ed Lee
DATE: June 21, 1993
SUB: **Yield reporting system for manufacturing area**

Hello

As most of you already know, I will be here for the next few months working on an enhanced yield-reporting and defect-tracking method for the manufacturing area. I've spent the last couple of weeks informally interviewing people and learning some of the ins and outs of the process. Now, its time to get down to business and start to get this thing done.

Initially, I have the idea to put together some sort of windows-based tool to access data from the floor. On the operator end, data on operations and yields would be entered via QIC or some other screen menu. On the other end (the output side), anything is possible. This is where I need your help.

I want to make sure that anything I do *will actually be useful*. Therefore, I need to know what kinds of data that you folks actually need and use. Please take a few minutes to fill out this survey. The questions here are rather open-ended, and there will probably be a wide range of answers (or -- maybe not). I can't promise to please everyone here, but knowing some of this information will put me on the right track.

The Survey

This survey should only take a couple of minutes. I will keep all answers to myself, so feel free to be frank and honest. Also, don't be afraid to get carried away and wordy, the more information I have, the better. I will be handing out a similar survey to the operators, since they will also have to live with the system.

APPENDIX F

Yield Reporting System Survey and Given to Engineers and Managers (continued...)

Name: _____

Process Responsibility: _____

- 1.) *How do you currently get information on yields for specific processes?*

- 2.) *Is the data that you find on the floor or elsewhere always in a format that you need? If not, how do you convert it into a usable format? (i.e. "I type information from process X into a Lotus spreadsheet at my desk and plot whatchamacallits v. beenie weenies.")*

- 3.) *If data on overall product yields were to be presented for any time horizon, what would be most useful to you? (weekly, daily, yearly, etc.....) How would this be different for specific process yields?*

- 4.) *Some people find first-pass yield information to be the most useful, while others like to see information on an overall end-pass basis. Which is most important to you? Why? If first-pass and rework information is important, how would you like to see it presented?*

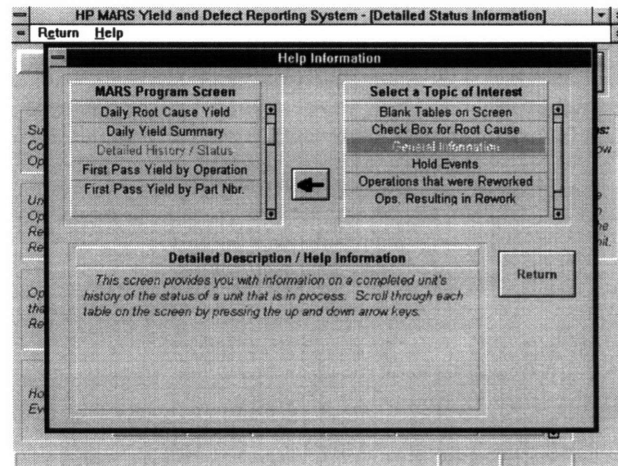
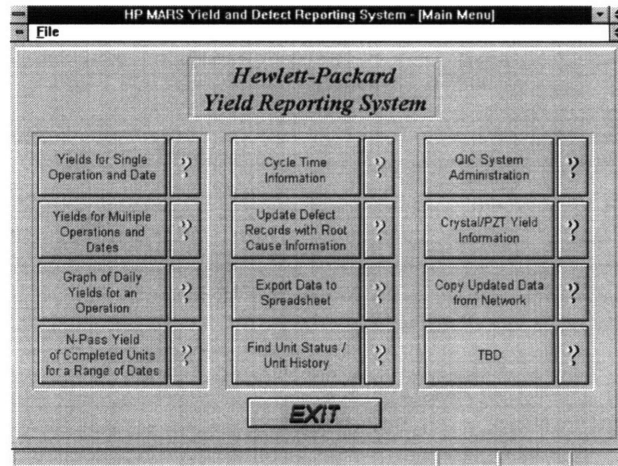
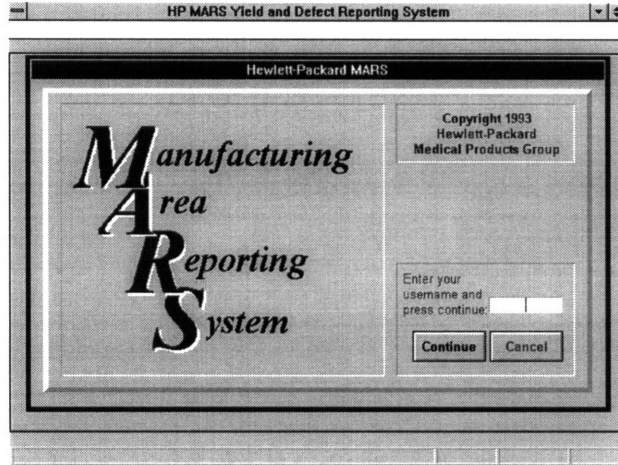
- 5.) *Often, products with defects triggered by one process have these defects found by other processes down the line. How is information on these "downstream defects" fed back to you for processes that concern you? How might you like it done differently?*

- 6.) *If a genie appeared out of a bottle and granted you three wishes for a computer-based system to report yields, what would they be?*

- 7.) *Other comments:*

APPENDIX G

MARS Screens



APPENDIX G

MARS Screens

HP MARS Yield and Defect Reporting System - [Main Menu]

File

Yields for Single Operation and Date

Choose Search Criteria

Part Numbers to include:

- All Part Numbers
- Specific Part

Enter Part No.:

Date: Operation:

Run Search Cancel

Production area to:

- Component Mfg.
- Wire and Test
- System Assembly
- Component Assembly
- Material Preparation

Include information on:

- New Parts
- Customer Exchanges
- Marketing Returns
- Promotional Parts
- Prototypes

HP MARS Yield and Defect Reporting System - [Yield Information for an Operation]

Search Return Report Pareto Help

Yield Information - Base Screen

Component Mfg.

Operation: Date:

New Marketing Return
 Customer Exchange
 Promotion Prototype

New Search Defect Pareto Print Report

Units Started		First Pass Information		N-Pass Information		Cycles
Unit Number	Part Number	Good Unit	Scrap or Rework	Good Unit	Scrap	Number of Passes
A5295	AB3113-3	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	1.00
A5297	AB3112-2	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	1.00
A5300	AB3116-6	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	2.00
A5303	AB3109-9	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	1.00
A5304	AB3108-8	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	1.00
A5312	AB3109-9	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	1.00
A5314	AB3115-5	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	1.00
Totals:		20	2	21	1	
Number of Starts: 22		Yield: 90.9 %		Yield: 95.5 %		Avg.: 1.09

Root Cause Yield Main Menu HELP ?

16 of 22 [PRIV:PNOPI.DB]

HP MARS Yield and Defect Reporting System - [Root Cause Yield Information]

Return Help

Root Cause Yield

Yield Summary

Number of Starts	Good Parts	Parts Scrapped for Related Defects	Root Cause Yield
22	21	1	95.5 %

Percent of Scraps with Root Cause Analysis:

Return to Last Screen HELP

Yield Detail: Units that started operation 40 on 1/10/93, went on to be scrapped for Cause Codes associated with that operation as shown in the table below.

Unit Number	Part Number	Operation Where Scrapped	Defect Root Cause	Date Unit was Scrapped
A5304	AB3108-8	50	4003 Cap too big	1/10/93

1 of 1 [PRIV:CAUSE1.DB]

APPENDIX G

MARS Screens

HP MARS Yield and Defect Reporting System - [Main Menu]

Yields for Multiple Operations and Dates

Search Criteria

Include Information on:

- New Parts
- Customer Exchanges
- Marketing Returns
- Promotional Parts
- Prototypes

Find information for completed parts and the following production stages:

- All Operations
- Single Operation

Enter Operation:

Yield Number of Interest:

- First Pass Yield
- N-Pass Yield
- Root Cause Yield

Production area to search:

- Component Mfg.
- Wire and Test
- System Assembly
- Component Assembly
- Material Preparation

Part Numbers to include:

- All Part Numbers
- Specific Part

Enter Part No:

Desired Date(s):

- Specific Date: Date:
- Range of Dates: From: To:

View Results by:

- Part Number
- Operation

HP MARS Yield and Defect Reporting System - [First Pass Yield Search Results b

Search Return Report Pareto Help

First Pass Yields by Part

Component Mfg. Product Types

- New
- Marketing Return
- Customer Exchange
- Promotional
- Prototype

Requested Dates

From: 2/1/93 To: 2/26/93

Display Option

- All Part Numbers
- List by Part Number

Part Number

- AB3101-1
- AB3102-2
- AB3103-3
- AB3104-4
- AB3105-5

Operation	Number of Starts	Good Units	Units Terminated or Needing Rework	First Pass Yield
2	61	54	7	88.5 %
8	54	45	9	83.3 %
12	48	39	9	81.3 %
35	47	43	4	91.5 %
40	44	41	3	93.2 %

1 of 16 [PRIV:PNDP2.DB]

HP MARS Yield and Defect Reporting System - [First Pass Search Results by Ope

Search Return Report Pareto Help

First Pass Yields by Operation

Component Mfg. Product Types

- New
- Marketing Return
- Customer Exchange
- Promotion
- Prototype

Requested Dates

From: 1/1/93 To: 2/15/93

Operation

- 2
- 8
- 9
- 10
- 12

Part Number	Number of Starts	Good Units	Units Terminated or Placed on Rework	First Pass Yield
AB3101-1	94	84	10	89.4 %
AB3103-3	100	98	2	98.0 %
AB3104-4	89	83	6	93.3 %
AB3105-5	99	92	7	92.9 %
AB3106-6	95	84	11	88.4 %
Summary:	477	441	36	92.5 %

1 of 22 [PRIV:OPPN2.DB]

APPENDIX G

MARS Screens

HP MARS Yield and Defect Reporting System - [N-Pass Search Results by Part Num]

Search Return Report Pareto Help

N-Pass Yields by Part Number

Component Mfg. Requested Dates From: 1/1/93 To: 1/10/93

Product Types:

- New
- Marketing Return
- Customer Exchange
- Promotion
- Prototype

New Search Main Menu Defect Pareto Print Report HELP ?

Display Option: All Part Number (selected) List by Part Number

Operation	Number of Starts	Good Units	Terminated Units	N-Pass Yield	Average Number of Passes
2	21	18	3	85.7 %	1.10
8	16	14	2	87.5 %	1.13
12	12	12		100.0 %	1.33
35	11	11		100.0 %	1.09
40	9	9		100.0 %	1.00
62	9	9		100.0 %	1.00

Part Number list: AB3101-1, AB3102-2, AB3103-3, AB3104-4, AB3105-5

1 of 16 [PRIV:PNOP2.DB]

HP MARS Yield and Defect Reporting System - [N-Pass Search Results by Operatio]

Search Return Report Pareto Help

N-Pass Yields by Operation

Component Mfg. Requested Dates From: 1/1/93 To: 1/31/93

Product Types:

- New
- Marketing Return
- Customer Exchange
- Promotion
- Prototype

New Search Main Menu Defect Pareto Print Report HELP

Operation list: 2, 8, 9, 10, 12

Part Number	Number of Starts	Good Units	Terminated Units	N-Pass Yield	Average Number of Passes
AB3101-1	62	55	7	88.7 %	1.13
AB3103-3	62	60	2	96.8 %	1.08
AB3104-4	60	55	5	91.7 %	1.10
AB3105-5	61	57	4	93.4 %	1.08
AB3106-6	69	61	8	88.4 %	1.06

Summary: 314 288 26 91.7 % 1.09

1 of 22 [PRIV:OPPN2.DB]

HP MARS Yield and Defect Reporting System - [Root Cause Yield Search Results b]

Search Return Report Pareto Help

Root Cause Yields by Part Number

Component Mfg. Requested Dates From: 1/1/93 To: 1/31/93

Product Types:

- New
- Marketing Return
- Customer Exchange
- Promotion
- Prototype

New Search Main Menu Defect Pareto Print Report HELP

Display Option: All Part Numbers (selected) List by Part Number

Part Number list: AB3101-1, AB3102-2, AB3103-3, AB3104-4, AB3105-5

Operation	Number of Starts	Good Units	Number of Terminants	Ultimate Yield
2	62	50	12	80.6 %
8	55	47	8	85.5 %
12	50	47	3	94.0 %
35	49	49		100.0 %
40	46	44	2	95.7 %
62	44	43	1	97.7 %
68	42	41	1	97.6 %

Percent of Scraps with Root Cause Analysis 0.3 %

1 of 16 [PRIV:PNOP2.DB]

APPENDIX G

MARS Screens

HP MARS Yield and Defect Reporting System - [Root Cause Yield Search Results b]

Search Return Report Pareto Help

Root Cause Yields by Operation

Component Mfg. Requested Dates From: 1/1/93 To: 1/31/93

Product Types:

- New
- Marketing Return
- Customer Exchange
- Promotion
- Prototype

New Search Main Menu
Defect Pareto
Print Report HELP ?

Operation	Part Number	Number of Starts	Good Units	Units Terminated or Placed on Rework	Root Cause Yield
2	AB3101-1	62	50	12	80.6 %
8	AB3103-3	62	53	9	85.5 %
9	AB3104-4	60	50	10	83.3 %
10	AB3105-5	61	50	11	82.0 %
	AB3106-6	69	54	15	78.3 %

Percent of Scraps with Root Cause Analysis: 0.3 %

Summary: 314 257 57 81.8 %

1 of 22 [-PRIV:OPPN2.DB]

HP MARS Yield and Defect Reporting System - [Pareto of Defects]

Return Help

Pareto of Defects by Operation

Component Mfg. Scrapped Units Rework Events % of Pareto based on actual Root Cause Analysis: 0.1 %

Return

Percent of Total

From: 1/1/93 To: 1/31/93

View Options:

- All Operations
- Each Operation

Operation: 9, 10, 15

Defect Description	Freq.
204 Chrome Missing from Bumper	7
802 Left Side Door Won't Open	9
203 Bumper Scatched	13
803 Right Side Door Won't Close	13
202 Bumper Dentad	16

5 of 22 [-PRIV:PARP2.DB]

HP MARS Yield and Defect Reporting System - [Pareto of Defects]

Return Help

Pareto of Defects by Part Number

Component Mfg. Scrapped Units Rework Events % of Pareto based on actual Root Cause Analysis: 0.0 %

Return

Percent of Total

From: 1/1/93 To: 1/10/93

View Options:

- All Part Numbers
- Each Part Number

Part Number: AB3101-1, AB3102-2, AB3103-3, AB3104-4

Defect Description	Freq.
1204 Door Handle Missing	5
201 Rust Spots in Bumper	5
203 Bumper Scatched	5
803 Right Side Door Won't Close	5

3 of 16 [-PRIV:PARPART2.DB]

APPENDIX G

MARS Screens

HP MARS Yield and Defect Reporting System - [Main Menu]

File

Enter Search Options

Choose Search Criteria

Production area to

- Component Mfg.
- Wire and Test
- System Assembly
- Component Assembly
- Material Preparation

Part Numbers to include:

- All Part Numbers
- Specific Part

Enter Part No. _____

Include information on:

- New Parts
- Customer Exchanges
- Marketing Returns
- Promotional Parts
- Prototypes

Desired Date(s):

From: _____ To: _____

Run Search Cancel

HP MARS Yield and Defect Reporting System - [Terminant Yield Information Summary]

Search Return Report Pareto Help

Terminant Yields

Component Mfg.

From: 1/1/93 To: 1/31/93

Product Types

- New
- Marketing Return
- Customer Exchange
- Promotion
- Prototype

New Search Main Menu

Defect Pareto Print Report HELP ?

View Options

- All Part Numbers
- Each Part Number

Yield Summary

Number of Units Shipped or Scrapped during Time Period: 61

Number of Good Units: 32

Number of Scrapped Units: 29

N-Pass Yield: 52.5 %

Part Number

- ASB101-1
- AB3102-2
- AB3103-3
- AB3104-4
- AB3105-5

1 of 16 [PRIV:PAR4.DB]

HP MARS Yield and Defect Reporting System - [Pareto of Defects]

Return Help

Pareto of Defects by Part Number

Component Mfg.

Scrapped Units Rework Events

% of Pareto based on actual Root Cause Analysis: 0.2 %

Return

From: 1/1/93 To: 1/31/93

Part Number

- AB3104-4
- AB3102-2
- AB3103-3
- AB3104-4

View Options:

- All Part Numbers
- Each Part Number

Percent of Total

Part Number	Percent of Total
204	4.0%
202	10.0%
201	10.0%
803	8.0%
1202	8.0%
804	8.0%
801	8.0%
1201	8.0%
802	4.0%
4004	4.0%
203	4.0%
1203	4.0%
8003	2.0%
8001	2.0%
8901	2.0%
8204	2.0%
8201	2.0%
4002	2.0%
3502	2.0%
1204	2.0%

Defect Description Freq.

1204	Door Handle Missing	1
3502	Paint Flaked in Hood	1
4002	Hood Won't Open	1
6201	Grills Damaged	1

1 of 16 [PRIV:PARPART4.DB]

APPENDIX G

MARS Screens

HP MARS Yield and Defect Reporting System - [Main Menu]

File

Graph of Daily Yields for an Operation

Graphing Criteria

Production area to search:

- Component Mfg.
- Wire and Test
- System Assembly
- Component Assembly
- Material Preparation

Yield Number of Interest:

- First Pass Yield
- Average Number of Passes
- N-Pass Yield
- Root Cause Yield

Part Numbers to include:

- All Part Numbers
- Specific Part

Include information on:

- New Parts
- Customer Exchanges
- Marketing Returns
- Promotional Parts
- Prototypes

Desired Date(s):

From:

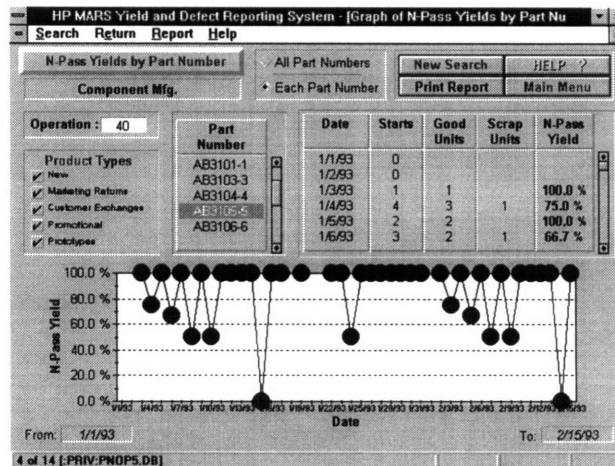
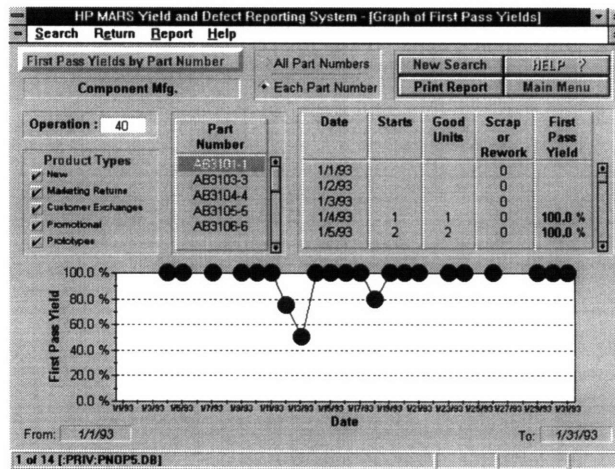
To:

Operation:

Run Search

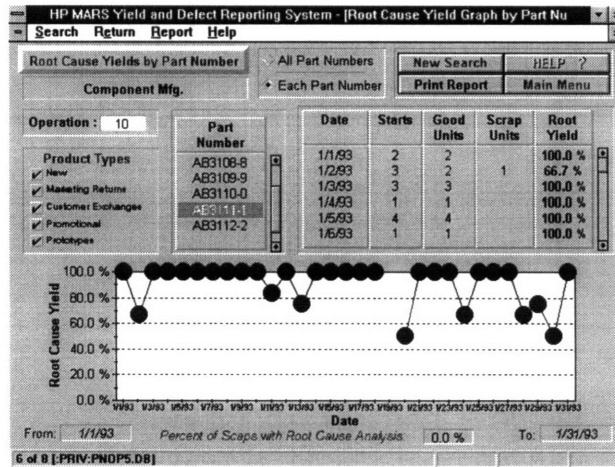
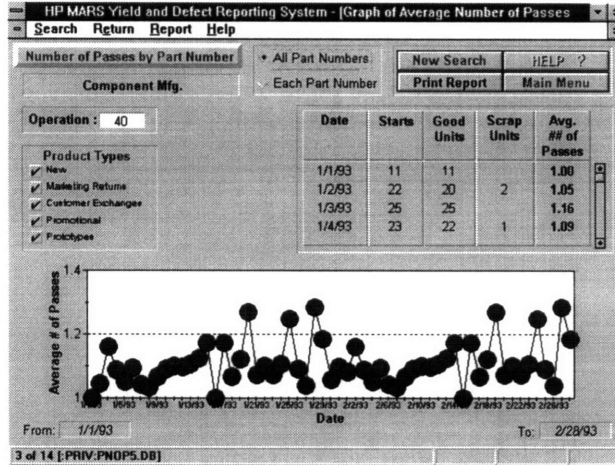
Cancel

EXIT



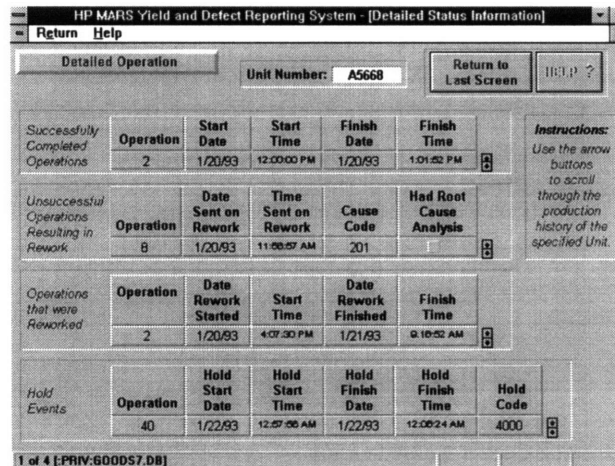
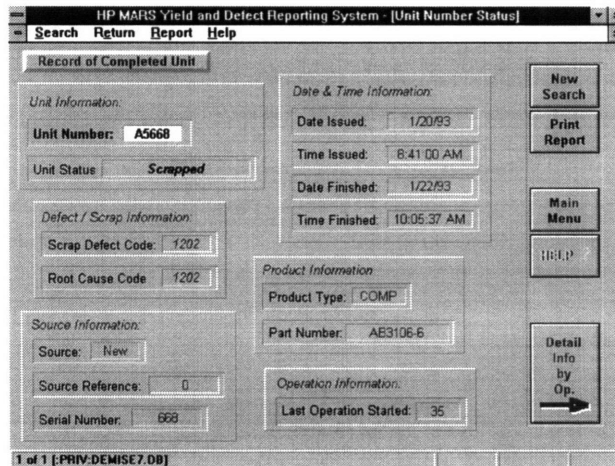
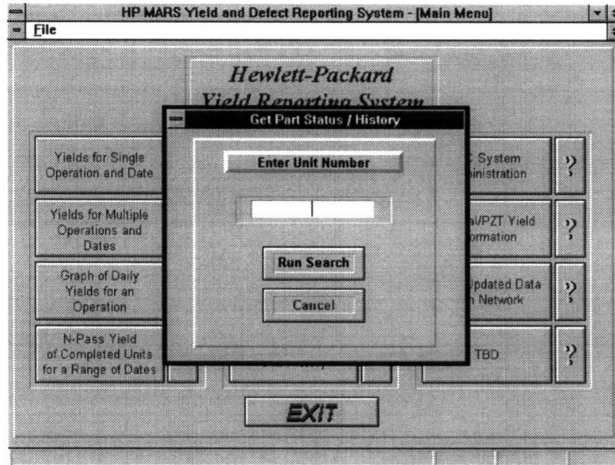
APPENDIX G

MARS Screens



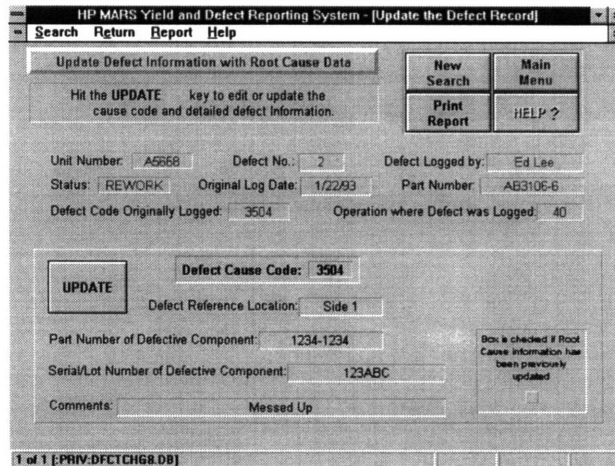
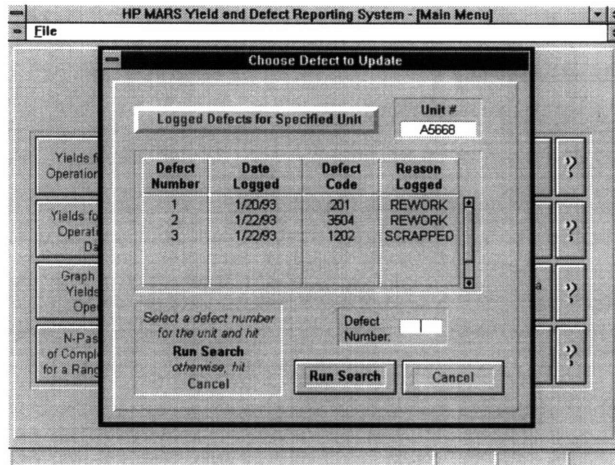
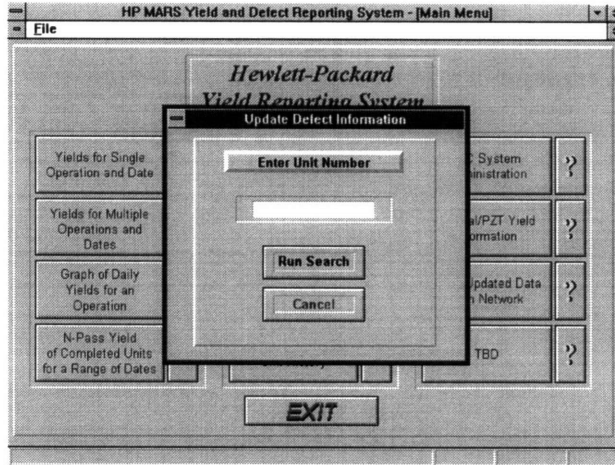
APPENDIX G

MARS Screens



APPENDIX G

MARS Screens



APPENDIX G

MARS Screens

HP MARS Yield and Defect Reporting System - [N-Pass Search Results by Part Num]

Search Return Report Pareto Help

N-Pass Yields by Part Number

Component Mfg. Product Types New Search Main Menu

Requester From: 1/1/93 To: 1/31/93

Display Option: All Part Numbers List by Part Number

Part Number

AB3101-1	%	1.13
AB3102-2	%	1.13
AB3103-3	%	1.26
AB3104-4	%	1.12
AB3105-5	%	1.07

40 46 44 2 95.7 % 1.07

62 44 43 1 97.7 % 1.07

Choose Pareto Options

Choose the type of defect event for the Pareto Chart.

Scrap Causes Only

Rework Causes Only

Both Scrap and Rework

Run Pareto Cancel

HP MARS Yield and Defect Reporting System - [Pareto of Defects]

Return Help

Pareto of Defects by Part Number

Component Mfg. Scrapped Units Rework Events % of Pareto based on actual Root Cause Analysis: 0.0 % Return

Date: 1/10/93 Op: 40

View Options: All Part Numbers Each Part Number

Percent of Total

Defect Cause Description	Freq.
4003 Cap too Big	1
801 Right Side Door Won't Open	1

1 of 2 [PRIV:PARPART1.DB]

APPENDIX G

MARS Screens

The screenshot shows the 'Export to Spreadsheet' dialog box. It has a title bar 'HP MARS Yield and Defect Reporting System - [Main Menu]' and a 'File' menu. The dialog is divided into several sections:

- Data Criteria:** A list of checkboxes under 'Include information on:'.
 - New Parts
 - Customer Exchanges
 - Marketing Returns
 - Promotional Parts
 - Prototypes
- Production area to search:** A list of radio buttons.
 - Component Mfg.
 - Wire and Test
 - System Assembly
 - Component Assembly
 - Material Preparation
- Find information for completed parts and the following production stages:** Radio buttons for 'All Operations' (selected) and 'Single Operation'. An 'Enter Operation' field is next to 'Single Operation'.
- Desired Date(s):** Radio buttons for 'Specific Date' and 'Range of Dates'. A 'Date' field is next to 'Specific Date'. 'From' and 'To' fields are next to 'Range of Dates'.
- Part Numbers to include:** Radio buttons for 'All Part Numbers' (selected) and 'Specific Part'. An 'Enter Part No.' field is next to 'Specific Part'.
- Buttons:** 'Continue' and 'Cancel' buttons are located at the bottom right.

The screenshot shows the 'Export Files to Spreadsheet' dialog box. It has a title bar 'HP MARS Yield and Defect Reporting System - [Main Menu]' and a 'File' menu. The dialog is divided into several sections:

- Spreadsheet Export:** A section with a title and a paragraph: 'Four tables of data will be exported to the directory that you specify. Please enter eight character filenames** for the tables, then hit the Export key.'
- Desired Export Format:** A list of radio buttons.
 - Quattro
 - Quattro Pro for Windows
 - Quattro Pro for DOS
 - Excel 3.0 / 4.0
 - Lotus 1.A
 - Lotus 2.x
- Export Directory:** A text field containing 'C:\PDXWINPRIVATE'.
- Table Selections:** Four radio buttons, each with a description and a label.
 - Table containing activities performed at each operation: **ACTIVITY**
 - Table containing a historical record for each unit number: **HISTORY**
 - Table containing a record for each defect or rework event performed: **DEFECTS**
 - Table containing a record for each part that is currently in-process: **UNITS**
- Buttons:** 'Export' and 'Cancel' buttons are located in the center.
- Footnote:** A small note at the bottom: '** do not include the extensions in the filenames that you enter. They are attached automatically.'