Optimizing the Operations Environment through the use of Manufacturing Techniques

by Victor Rozek
and
Sheila Dummer;
Hewlett-Packard

Two percent of sales, a loose standard for estimating data processing budgets, can easily run into the millions for a mid sized company. The business end of the budget is generally reserved for massive hardware purchases. Applications software packages and the escalating salaries of the programming staff. Almost as an afterthought, what can not be spent elsewhere is allocated to the support of computer operations.

Computer operations is the Rodney Dangerfield of data processing. Operators are generally entry level personnel, marginally trained and poorly compensated. Often working alone during off-shift hours. They are denied access to the high priced expertise of those who cause most of their problems. It is fashionable to blame them for everything from sloppy programming to hardware failures.

The theory is that anyone can run a program. Quite so, but even the most demanding gentle reader would have difficulty running 50 to 200 programs while keeping track of schedule requirements, sequence, job dependency, prompts, number of copies, special forms, restart and recovery procedures and media requirements. Yet that is precisely what many operators are asked to do with little more than a 3X5 card with scribbled instructions to guide them.

Yet so much of the success of a data processing department depends on the efficiency of its operations staff. The finest programming efforts are rendered meaningless if the G/L is closed before the receivables are posted; if people can't get their work done because there aren't enough copies of a critical report, if manufacturing must quesstimate requirements because an 18 hour MRP blew up and there were insufficient restart instructions.

This paper is submitted in the hope of providing help for the beleaguered operations staff. Areas of concern and neglect will be defined and solutions presented based on a model of manufacturing techniques. We will attempt to identify methods for optimizing throughput in a batch environnment, and providing timely and accurate output while generating a minimum of friction between the applications, operations and user communities.

Several assumptions are made: notably that we are all pregnant with Packard and therefore one or more HP3000 systems are being used in a business environment. Batch processing is defined as a combination of updates and reports from multiple data bases, typically run during evenings and weekends. During batch processing, time share users are at a minimum. Further, we strongly suspect that the HP3000 was designed to be a multiprocessing system, optimized for the performance of multiple simultaneous tasks - in spite of your experience to the contrary.

The first step in developing a manufacturing to data processing analogy is to examine briefly the functional organization of each group. Figure 1 illustrates a typical manufacturing organization and shows a comparable functional organization for a data processing department. Note that these are functional organizations; in small data processing groups several functions may be performed by the same individual. Overlooking any function is, however, a common cause of ineffective operations.

Using the data processing to manufacturing analogy, the first concern is capacity planning. Generally a sales force forecasts sales and on the basis of these projections, production planning estimates manpower, equipment and material requirements. In our model, business systems analysis, with input from the user community, forecast hardware requirements including disc space, CPU processing time and printed output, based on projected growth in existing application systems and new applications under development. The operations and/or technical support group plan new hardware purchases and schedule installation and training.

Formalizing this process and matching it to the company's budget insures that aquisition of additional hardware resources coincides with the timing and scope of the requirements. Form 1 in the attached forms appendix is a suggested questionnaire that can be used to solicit input from the applications programming group to the operations group.

Production, of course, is the name of the game both in manufacturing and computer operations. Traditionally, R&D designs a product which manufacturing engineers translate to production specs in a prescribed standard format. Likewise, business systems analysts, responding to user needs, develops a system design. That design is submitted to systems and applications programmers who develop the system, including batch jobs, according to department standards. When complete, operations scheduling or a separate quality assurance group (sometimes a maintenance programming group) reviews and accepts the system and supporting documentation.

In the manufacturing environment, production control accepts engineering designs and develops assembly plans and production specs. Materials and manpower requirements are identified and scheduled. Production then assembles the product according to specs. The process is gradually refined and statistical data is collected to provide feedback to R&D and production control.

Likewise, operations schedules batch processing requirements which will determine staffing levels and needed resources such as run time, disc space, paper, special forms and tapes. As jobs are processed, operations monitors console pages and logfiles to compile statistics on run times, frequency and cause of job failures. These statistics are forwarded to applications for resolution of processing problems.

Conversely, the results of the processing cycle are reviewed by data control personnel who provide statistical feedback to operators on missed schedules. These personnel typically distribution staff, tape librarians or I/O control staff provide a system of checks and balances that insures adherence to processing procedures and documentation, as well as providing continuous validation of those procedures and documentation. Schedule volumes or concurrent job mix is adjusted to available processing time. Similarly, the manufacturing Q.A. function reviews production, providing feedback used to improve both the product and the manufacturing process.

Finally, the product is packaged and shipped to the customer. Defective or damaged product is returned and repaired, and repair cause and frequency statistics are kept to improve future product reliability. Computer operations distributes batch output to the users. If inaccuracies exist, either in the form of program bugs which cause job aborts or unsound data, applications makes needed modifications. Feedback from operators can also be used to optimize future design.

Understanding the composition and function of the model is only the first step. Optimizing it is the critical second step. Perhaps the single most important factor in successfully controlling computer operations is the establishment and enforcement of standards. This has long been recognized in manufacturing where standard "assembly line" techniques have been used for over 150 years. Beyond providing a continuity, adequate standards can minimize the need for operator training while maximizing flexibility. The three major candidates for standardization; jobs, documentation and schedules.

A caution; to effectively develop and maintain standards will require the cooperation volitional or otherwise, of the applications and operations groups. That may prove difficult. Programmers, being the creative, free spirited, willfull, overpaid group that they are, often resent the imposition of standards. Management may have to flex managerial muscle to ensure a happy compliance.

Jobs names can be standardized to identify production jobs, updates, reports and the specific applications group to which the job belongs. It can be as generic or specific as site requirements dictate as long as the naming convention is observed by all. Keeping the convention simplistic is also esential since a complex naming standard can easily degenerate into no standard at all.

Input and output priorities should be standardized to allow critical jobs first consideration when the job queue is full and there are several jobs waiting to be processed. The more critical the job, the higher the inpri. Outpri can be used to identify forms type. An outpri of 9, for example, can indicate single part paper, while an outpri of 12 may indicate three part paper, and so on. This will allow the operator to print all available output of a certain type before changing paper, rather than constantly changing paper to accomodate the next available spoolfile. Anyone who has ever received four additional copies of a report they did not need, knows that relying on operator memory is risky at best. Enforcing use of the forms message capability for all special forms also insures that purchase orders do not print on sales order forms.

Tellop formats for tape mounts or other operator interface should be visually imposing to instantly alert the operator. That failing, a few well placed control G's can disturb even the most determined grave shift slumber. Tape names should have some meaning. Avoid generics like "DBSTORE" and "T". If 20 tapes are all labelled DBSTORE even the best operator is playing Russian Roulette trying to correctly mount and identify each one.

Standardize both the format and content of the tellops and insure that all information necessary for the operator to adequately label all tapes is presented on the console. (You to of course use preprinted tape labels either of the commercially available sort or stock blank labels printed with your very own 10 line BASIC program.) Tape labelling data sent to the console should include the name of the job creating the tape, the tape name (as identified in the formal designator defining the file), the retention period, the usage (should the write ring be in or out?), the BPI and a brief description of the contents (keep it short, tape labels are little tiny things!). Other useful information includes the date, the operator, the system (for a multimachine environment) and the MPE release. (Ever try to read a 2 year old store tape with a different blocking factor?)

Standardizing output file names helps too. It is difficult to sort out twenty reports with the useful title of "LP". A few well placed file equations can work miracles. Use comments liberally. Functions and restart points should be clearly commented. Job standardization can be a tremendous help to operations. Take full advantage of it.

Documentation standardization is equally helpful. It simply means that format should be the same for all jobs in allsystems. One of the problems with documentation is that many people do not know how to use it. If it takes more than a few seconds to locate the desired information, they either ask someone for the answer or simply give up. Getting accustomed to using one style of documentation creates a comfort level when using new and old documentation alike. People understand how the data is organized, and what information they can expect to find.

Documentation should include diagrams to illustrate job flow. Restart and recovery instruction should be clearly defined for each job, and those jobs which can not be restarted without programmer intervention should be identified. Detailed documentation for each job should include: the system on which it should be run (if multiple systems are used), the name of the programmer responsible for maintaining the job, the jobname.group.account where the production version is located, the job function, i.e. report, update etc., a short description of what the job does, any tape, microfiche or page print requirements, estimated run time, output and sector requirements, whether the job is critical or not, and whether the output is confidential, the schedule it runs on and any jobs or events that must preceed it.

One technique for standardizing documentation is to use forms. Forms insure that information is presented in a standard format, that only necessary data is provided and, conversely that no essential data is omitted. Use of forms creates easily updated documentation since operator manuals can be kept current by simply substituting a current form for an out of date one. Forms also provide excellent input documents for automated documentations and scheduling system. (See form samples in Appendix.) An automated system, whether homegrown or purchased provides strong benefits for the operations group struggling to keep track of hundreds of jobs. It should be considered essential in a multimachine shop.

The last question, once a standard documentation process is created, how do you maintain it? How do you quarantee that the programmer caught up in the thrill of the creative process comes to earth long enough to tell operations how to showcase his masterpiece to best advantage? Stockades, whipping posts, and public executions are one method. A protected library is a more humane alternative. The library is a group or an account to which only operations has write and save access. Read and/or execute access may be globally available. All production jobstreams are transferred to this account by a designated individual or group. The scheduler, I/O control or Quality Assurance groups are natural choices. Operators execute jobs from this library only and jobs are transferred to the library only after receipt of the appropriate documentation. This provides an easy, fairly administered, system to "remind" all concerned to keep documentation up to date. The only secret to maintaining such a library is enforcing a "no exceptions " policy. Any job requiring any form of operator attention--be it to stream it, alter its priority, mount a tape or print the output should come from the library. Take note; systems programmers and technical support personnel tend to try to "beat the system" more than applications programmers. A few violators can make the whole exercise meaningless so don't allow exceptions.

With such detailed documentation of each job, the operator should be able to handle any contingency, from rescheduling production due to prolonged system failure, to adjusting copy counts. Of course not all jobs run each evening. Standardizing schedules will alleviate the last minute scramble to identify which jobs genuinely qualify for production.

Scheduling: the art of detetermining which jobs run on which days and what resources (disc space, tapes, paper, special forms, DS line availability) are required. A good scheduler is a pearl beyond price; part wizard, part puzzle master, a mythological hero who can unravel the complexities of the Gordian knot before breakfast (or score millions of points on Space Invaders) and thinks nothing of planning 500 jobs running 8 at a time on each of 3 systems. Accurate scheduling for other than the smallest operations is impossible without automated assistance. A good scheduling system is linked to a strong documentation system and allows planning on a daily, weekly, monthly, quarterly and annual basis. However even an MPE file maintained in the Editor and printed with a simple report program can provide adequate automation. Schedules should be printed in a standard format for the operators and should include references to any necessary documentation. A section for operator comments and feedback helps answer the all important question "what ran last night?" Log file extract programs can provide an automated feedback process for the scheduler providing run time statistics, average lines of output and other essential detail down to the process level if needed. Again the contributed library is a good source for programs to read and report log files. Experiment with available programs and see what meets your needs. Remember too that operations is a dynamic environment. The program that does exactly what you need today may be extraneous overhead six months from now when your needs change. Review and refine the statistical process just as statisticians do for a production line. Use the logfiles to help focus on your current needs not just to produce reports that fill file cabinets and aren't used.

A final word about standards; the best standards are simple, are written down for general reference, and are enforced by automated means. A computer is impartial and consistent; a computer is never uneasy or embarrassed when asked to serve as a review or enforcement mechanism for coworkers; and a computer can't be bribed to make an exception "just this once."

A second manufacturing technique that can be used to optimize the operations environment is the elimination of repetitive manual labor. On the assembly line this means using everything from power tools to robots to eliminate the necessity for human intervention. This reduces manpower requirements and minimizes the human error factor. For the operations environment this means eliminating operator involvement in streaming jobs, manipulating spoolfiles or editing job files. Standards contribute to this, particularly a standardized outpri/ outclass/ jobfence scheme that reduces spoolfile manipulation.

Another technique is to coordinate creation of "master" jobs with the scheduling function. These are jobs with no function other than initiating sets of jobs according to the prescribed schedule for that evening. Use of JCW checking to trap program and job aborts and to control sequence requirements can significantly reduce operator involvement in job processing and control. Let each job start its successors based on conditions trapped by JCW. These conditions can also contribute to maximizing throughput by reducing system "dead" time, that is the time lost waiting for the operator to initiate the next step in a sequence of jobs. The contributed library is a good source for programs to help speed throughput and reduce wasted "dead" time. Remembering that the HP3000 is a multiprocessing system is key to maximizing throughput. A Series III can process 5-7 jobs simultaneously; a Series 64 10-12 jobs. It may be necessary to experiment in your environment to find the optimum job mix. To do this it is first essential to create jobs that can run simultaneously. Avoid such limiting processing techniques as exclusive file or data base access, clustering simultaneous events (several reports for instance) in one job rather than many.

In the same vein use jobstreams and UDC's wherever possible to streamline routine operator functions. These can include UDC's to set system paramenters (like jobfence, outfence, jobpri and limit) at different points during the day or to control devices (open DS lines, spool printers) after system restarts. Jobstreams can be used to perform routine diagnostic and maintenance functions like running MEMLOGAN and FREE2 daily or performing a weekly disc condense. Techniques like these that create a user friendly environment are a routine part of the implementation of any application system. Operations however is like the proverbial cobblers children who have no shoes, is generally the last place where normal good data processing practices are applied. Review your operations environment. Look at it as a connected system. If all the segments are preforming to your expectations guard your operations manager well; if not consider whether the solution lies in changing the approach not the people.

How much improvement can be expected from implementing these techniques? Well, like everything in the HP3000 world, that depends on a number of factors. Things like how big is your installation, how complex are the applications and how unstructured is the environment when you start? In a multiple machine environment (3 or more systems) clear definition of the scheduling and support functions and implementation of standards can produce throughput improvements of 30% or more. More significant is the reduction in the "hassle" factor. This is the level of stress created when programmers are called in the middle of the night, when users complain constantly about systems constantly down or output that is late or incmplete everyone's life (especially the operations manager's) is very stressed. Reducing this element decrease turnover increases communications and teamwork and creates a professional data processing environment. This is an immeasurable benefit; the most significant one that can be achieved!

APPENDIX

Form 1 - Capacity Input

Form 2 - Tape Label

Form 3 - Job Documentation

Form 4 - Operations Procedure

Form 5 - Distribution

Form 6 - Special Forms Print Documentation

Form 7 - Production Turnover (Scheduler's Instructions)

Form 10 - DataBase Capacity Change (Adager Request)

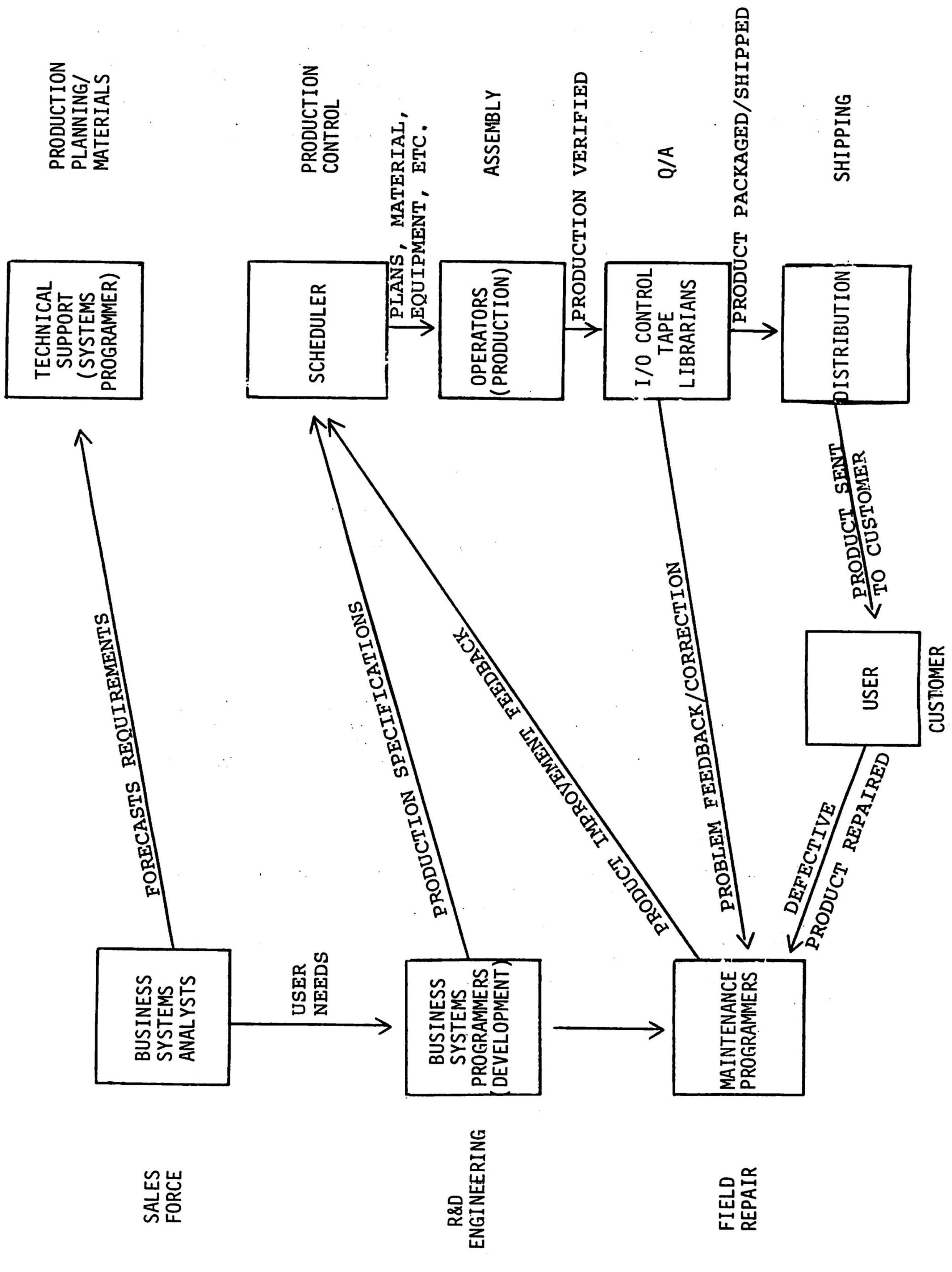
Biographical Sketch

Victor Rozek

Educated at San Francisco State University, Mr. Rozek has been involved in computer operations for the past eight years. He has held management positions with SILTEC Coporation, a California based silicon manufacturer, and Smith-Kline Clinical Labs, He is presently Data Processing Operations Manager for ITT Qume Corporation in San Jose, California.

Sheila Dummer

Ms. Dummer received a B.S. from the University of San Francisco and continued with graduate studies in the MBA program at that university. She has worked in data processing operations for 15 years holding management positions in computer operations, systems programming and applications system development with Wells Fargo Bank and ITT Qume Corporation. She has consulted in the area of operations optimization for the HP3000 for 5 years and has worked with the HP3000 since 1973. Ms. Dummer is currently Computer Operations Manager for the CBX division of ROLM Corporation.


PRODUCTION PLANNING/ MATERIALS
PRODUCTION CONTROL
ASSEMBLY
Q/A
SHIPPING
TECHNICAL SUPPORT (SYSTEMS PROGRAMMER)
SCHEDULER
PLANS, MATERIAL, EQUIPMENT, ETC.
OPERATORS (PRODUCTION)
PRODUCTION VERIFIED
I/O CONTROL TAPE LIBRARIANS
PRODUCT PACKAGED/SHIPPED
DISTRIBUTION
PRODUCT SENT TO CUSTOMER
USER
CUSTOMER
FORECASTS REQUIREMENTS
PRODUCTION SPECIFICATIONS
PRODUCT IMPROVEMENT FEEDBACK
PROBLEM FEEDBACK/CORRECTION
DEFECTIVE PRODUCT REPAIRED
BUSINESS SYSTEMS ANALYSTS
USER NEEDS
BUSINESS SYSTEMS PROGRAMMERS (DEVELOPMENT)
MAINTENANCE PROGRAMMERS
SALES FORCE
R&D ENGINEERING
FIELD REPAIR