Management: Key to Successful Systems Implementation

Gary A. Langenwalter
Manager, MIS
Faultless Division
Bliss & Laughlin Industries
Evansville, Indiana

When I arrived at Faultless four years ago, the new on-line Order Entry system was supposed to be completely operational. I found a completed general design, some detail design, and 10 programs coded. The hardware vendor (not HP) had promised Faultless management that they would contribute one person for one year, we would do likewise, and the result would be a state-of-the-art order entry system. We finished 1½ years late, with an investment of 6+ years of effort. We are currently replacing our old hardware with an HP3000, and replacing all our software. This conversion was scheduled to take 14 months, finishing November 30, 1981. Our best current projection is August 1982. In all fairness, I must mention that the Master Scheduling package that we bought three years ago was installed on time, under budget, and it met our expectations.

We at Faultless are not alone. Consider the following three disasters, all of which occurred in Fortune 500 companies in 1980:

> "A major industrial products company discovers one and a half months before the installation date for a computer system that a $15 million effort to convert from one manufacturer to another is in trouble, and installation must be delayed a year. Eighteen months later, the changeover has still not taken place.
>
> "A large consumer products company budgets $250,000 for a new computer-based personnel information system to be ready in nine months. Two years later, $2.5 million has been spent, and an estimated $3.6 million more is needed to complete the job. The company has to stop the project.
>
> "A sizable financial institution slips $1.5 million over budget and 12 months behind on the development of programs for a new financial systems package, vital for day-to-day functioning of one of its major operating groups. Once the system is finally installed, average transaction response times are much longer than expected." (McFarlan, p. 142)

Ollie Wight, the leading consultant in the manufacturing systems field, estimates that there are fewer than 25 "Class A" MRP users in the country! That number compares poorly to the multiple thousands of companies that have tried to implement manufacturing systems, each with the intent to succeed. We are one of the "thousands"; we are working to become "Class A."

The major risks of systems implementation can be categorized as follows:

1. Failure to obtain all, or even any, of the anticipated benefits.
2. Costs of implementation that vastly exceed planned levels.
3. Time for implementation that is much greater than expected.
4. Technical performance of the resulting systems that turns out to be significantly below expectation.
5. Incompatibility of the system with the selected hardware and software. (McFarlan, p. 143)

Three factors that determine the degree of risk are listed below:

1. Project size. The larger the size, the greater the risk. Size is also relative — a $500,000 project has much greater risk for a $20,000,000 company with a 3 person MIS staff that has never installed anything of its size, than for a $200,000,000 company with 20 programmer/analysts.
2. Experience with the technology. Unfamiliarity with the computer in question, or its operating system, or database, or TP monitor and terminals increases the risk.
3. Project structure. Having clearly defined inputs and outputs, which all users agree upon beforehand substantially reduces the project risk. I have not yet seen this, but it is theoretically possible. Conversely, when people are still changing basic systems functions and designs midway through the project, that project is doomed to overrun both temporal and financial budgets. The military is particularly adept at this (aided and abetted by the contracters).

My current experience, plus my previous background as an educator and consultant with a major DP hardware vendor, support the hypothesis that forms the basis for this talk:

HYPOTHESIS

The single factor most responsible for success or failure of system implementation is management. Good management requires identification and minimization of risk of failure, plus continual execution of the three basic management principles: Planning, Organizing, and Controlling.

The implementation of a system will be successful if, and only if, it meets three basic goals which are the converse of the risks listed above:

1. On-time completion.
2. On-budget completion.
3. The completed and installed system must meet both its specifications, and the users' expectations.

Let us review in some detail how each of the basic management techniques can be used to insure successful systems implementation.

PLANNING

Perhaps the best way to approach the topic of planning is with a cursory overview of the techniques available. Both PERT charts (or CPM charts, or "Bubble charts") and bar charts have been widely used for years. Appendix B includes a sample of each. In general, computer programs are a tremendous help in handling complex PERT charts, and recalculating critical paths.

Time estimating is perhaps the biggest stumbling block to proper systems implementation time and cost projections. Various articles suggest that each person on a project be scheduled at only 70% efficiency, and that one should allow 2-3 weeks for a user decision. My own personal experience indicates that one should allow 1-2 months for vendor feedback (to an RFP, for example), and for scheduling vendor presentations and reference visits. Also, if a person is managing others, 20% of his time should be allotted for each person managed, subject to the discretion of the estimator. Finally, an estimator needs to allow "Contingency time" of 20-200%, depending on the tightness of the other estimates, and the degree of risk inherent in the project — the contingency factor should increase proportionally with the risk.

Now, down to the actual planning itself. In my opinion, the only intelligent way to implement a large system is to break it into four phases, with management, the users, and MIS mutually agreeing to the functions, cost, benefits, and time estimates at the end of each phase. This minimizes the risks involved, and maximizes the probability that the user department will implement the finished product successfully. We will examine each phase below.

1. Initiation Phase

This phase includes the preliminary survey, a rough estimate of potential costs and benefits, and the selection of the alternative perceived to be the most attractive (make vs. buy, Vendor A vs. Vendor B, etc.). It culminates in a presentation to top management of the Systems Proposal written jointly by the user department manager and the MIS Manager. If top management approves, the system implementation enters phase 2. If not, it can be reworked or dropped, with minimal expenditures of resources to date.

This phase is the most important of all; it creates the basic expectations of system functionality in the minds of users and management. It should be noted that the basic system functions are defined by the person who will use them in his daily work, not by the MIS department representative. "Systems are tools for the manager, not toys for the technician." (Wight, p. vii)

Some of the topics which are covered in the Systems Proposal (or Management Overview) are management summary, major system benefits, economic justification, and schedule. Appendix A1 contains a more comprehensive list of topics included in the Systems Proposal.

One other topic which needs consideration throughout the implementation of a new system is the fear of change of the part of some people in the company. Some will be afraid that they will lose their jobs; others that they will not be able to measure up to the new expectations; and still others that they will lose their status with their peer groups, and/or that their work groups will be reorganized. These fears, unless addressed, can result in passive or active resistance to the new system on the part of the people whose daily enthusiastic cooperation is an absolute requirement. They must, therefore, be actively addressed and overcome.

2. Analysis Phase

This phase starts with a study which examines in greater detail all major assumptions and promises of the original proposal. Greater attention must be paid to any area that includes major uncertainty (response times with the particular hardware, application, and database under consideration, for example). Cost and benefits estimates are updated with the new information. My experience indicates that costs almost invariably increase, and benefits almost equally invariably decrease. Finally, the MIS department writes the Functional Specifications for the proposed system, and has them approved by the user department(s) affected. After they all agree, they jointly present them to the Steering Committee, with updated costs and benefits. If management approves, the project continues; if not, it is either discontinued, or revised. At this stage still, there has not been a major expenditure of corporate resources.

The Functional Specifications (or General Design) document can include the major logic chart, proposed input and output layouts, a training plan, future capabilities, and a contingency plan, to name but a few of the many topics. A more complete list appears as Appendix A2.

3. Design Phase

This phase defines how the system will be built. It is finished upon the completion of two major documents: the Design Specifications (or Detail Design), and the User's Manual.

Some topics that the Detail Design Specifications include are a detailed system flowchart, with input and output defined for each program, security, detailed program functions, specifications, and logic, and a detailed project implementation schedule. Appendix A3 contains a more exhaustive list.

One item that must be covered in appropriate detail is the Contingency Plan. All hardware, even HP's, and all software, even Faultless', will eventually fail. Such an event cannot be allowed to totally stop a critical department from functioning.

The User's Manual includes pictures and descriptions of all input and output screens, and reports, with explanations of all fields — what they mean, and how to change their contents, if appropriate. It also includes operating instructions (how to sign on to the system, what to do in case of problems, etc.). It must be written in language that the person in the functional department will easily understand.

These two major documents, plus Contingency Plan, are jointly presented by the user department manager and the MIS manager to the Steering Committee, with the re-revised cost/benefits data. If management approves, the system enters the final phase of implementation. If not, the minimum resources possible have been expended thus far; the project can be either revised or dropped. At this stage, all parties involved will have agreed on the details of the new system; there should be no "surprises" from here on out. There should be no reservations about technical capabilities, or about what the system will do.

4. Construction Phase

This phase is the one that includes the actual programming, testing, and documentation. In a well-managed project, more than 50% of the time should already have been spent designing. This minimizes changes, revisions, etc. that are the bane of efficient and effective systems. Let us discuss each subphase independently.

Programming is a complex enough topic that it warrants books, talks at this convention, and week-long training courses. Let me outline my views briefly, and then continue with the subject at hand. All programming should be top-down, structured, and modular. Each program or module must be tested and documented as soon as it is completed. It is then, and only then, that it can be included in the account that contains completed programs.

I will knowingly raise a controversy by suggesting that users should design their own screens (with V/3000, where applicable), and write their own reports (we are using REX for that purpose). To me, the data belongs to the user. Assuming that he understands the contents and implications of the numbers that exist in the database (and he should, for in most cases we hold him responsible for their accuracy), then he should be given the tools to generate the reports and inquiries that will allow him to manage his portion of corporate resources optimally. In other words, I refuse to perpetrate an "IBM" (International Brotherhood of Magicians) image with regard to my department.

The Systems Manual is a major document. It needs to follow predetermined specifications and formats, and, more importantly, must be updated throughout the life cycle of the system. There are very few things more dangerous than a slightly outdated Systems Manual in the hands of a programmer who is trying to maintain a system.

The Operations Manual is a must, whether your MIS department has a formal operations group or not. This document tells the operator how to run the batch jobs, back up the system, recover in the event of failure, where to send the output, etc. It defines expectations. If there is no formal document, the person who normally runs the job is generally the only person in the company with that information. The financial risk that represents to a company increases with the importance of the application (for example, weekly payroll).

Training cannot be overemphasized. The responsibility for training users lies with the Project Manager (the user department manager) rather than with MIS, because the head trainer becomes the person who knows the application better than anyone else in the organization. In smaller organizations, the Project Manager will train users directly; in large organizations, he will train other managers, who will then train their own people.

Training can and must commence as soon as the first few programs are finished. After the Project Team has trained itself, it is time to start familiarizing other personnel with the screens and reports they will be using soon. These people can often suggest invaluable improvements, some of which take almost no time to incorporate. The ones that involve much time must be prioritized, and approved by the Steering Committee prior to inclusion. The end users will also spot program flaws that escaped everybody else.

All user training and all program testing, except volume and response time testing, must be performed on a small test database, preferably one distilled from your real live database. My user personnel respond much more favorably to reports which include casters that they do to reports which feature bicycles.

Training is the one place that most people grossly underestimate the time and resources required for a proper implementation. Most people also underestimate the numbers of people that must be trained, and perhaps even educated. Training materials can be acquired from the vendor, if the software is purchased, and from video-tape training companies such as ASI and Deltak.

One has three choices for final testing: Parallel testing (which works well for financial systems, for example), Pilot testing (which can be used for some manufacturing systems implementations), and None (which I cannot recommend; the only cold turkey that I like is that which is left over from Thanksgiving dinner).

Final testing also quickly unearths any latent run time or response time problems. Although painful, and embarrassing, it is better to discover those problems at this stage than to try to squeeze 25 hours of processing into a 24 hour day after the old system has been cut off!

After the final testing is complete, one faces the actual conversion. Although this sounds simple ("Just take the old data and load it into the new database."), it can be most complex. Each type of data to be converted must be examined. Each outstanding piece of paper must be considered (Do we leave it there? Replace it? How do we find them all? What about the ones we miss?). To illustrate the complexity of such a task, consider that it took us at Faultless the entire Labor Day weekend, running around the clock, to cut our MRP database over from our other (non-HP) computer to the Series III. The process involved over 30 steps. The process and programs were so complex that we ran test runs on the conversion programs themselves several times.

ORGANIZING

Since the most important person in an implementation effort is the Project Manager, let us start by briefly defining his (her) attributes and responsibilities.

The Project Manager, in my opinion, must come from the department most affected by the project (that is, the one that will gain the most if it succeeds, and lose the most if it does not). It should be the person who will manage that function on a day-to-day basis after the system is successfully installed. The MIS Manager should be Assistant Project Manager, to insure that what the user wants is technically feasible. On a major project, the Project Manager position involves a full-time effort, especially when training commences. I know that in the "Real World," those people are often totally busy just keeping the company running on a day-to-day basis. But nobody else has the intimate knowledge of how that department really functions on a daily basis that is required for successful design and implementation of the new system. Faultless top management backs this philosophy 100%, by saying that if a department is not interested enough to furnish a Project Manager, the project will not commence.

The Project Manager is responsible for writing the functional specifications at the commencement of the project. They form the basis for all subsequent development. In my opinion, if a company does not have the time to write its own Functional Specification, (or RFP), and feels that it must hire a consulting firm to give birth to a 250-page document, that company has no business trying to implement any system that arises from that document, because it will not be "their" system. That system stands, in my opinion, a better than 90% chance of failure.

The Project Manager must plan, organize, and control (in other words, Manage) the day-to-day efforts of the project. He must continually check to make sure that detailed designs will meet the needs of his (and others') departments. He must monitor progress to schedule, and adjust the schedule to the realities that intrude on the best plans. He must control requests for changes by sitting on most of them, and presenting the few worthy ones to the Steering Committee. He must chair the Project team at its weekly meetings, and the Steering Committee at its bi-weekly meetings. As mentioned earlier, the Project Manager is also the head trainer, and trains either user personnel directly, or their managers who in turn train their subordinates).

The Project Team is comprised of the Project Manager (Chairman), MIS Manager (Assistant Chairman), managers of all departments affected, and the analysts and programmers assigned to the project. It is responsible for resolving differences of opinion that do not involve policy or fundamental operating philosophies, recommending policy and operations changes to the Steering Committee, ensuring that the project progresses as scheduled and results in the benefits promised, and prioritizing the myriad requests for changes, modifications, enhancements, etc. that occur in such projects. It must also monitor the creation and installation of internal controls, and contingency plans.

To be effective, the Project Team needs to meet weekly (a standing meeting time and place is usually appropriate). They need to keep a formal "Problem List," with the status of each problem, including its final resolution and date. This will ensure that a problem does not get ignored until it becomes extremely costly to resolve. The Project Team must send minutes of its meetings to the Steering Committee, with the Outstanding Problem sheet attached, annotated to show how each problem will be resolved. Finally, the members of the Project Team must be the ones who train on the new system first, and best. They will be assisting their subordinates to use the system correctly; they need to understand well how it works. They also need to know the inner workings of the new system so that the many decisions that must be made during an implementation will be the best possible.

The Steering Committee is comprised of the Project Manager (Chairman), MIS Manager (Assistant Chairman), the top executive of each department affected ("mahogany row," if you will), and the person to whom those executives report (the "corner office"). This committee should meet bi-weekly (more frequently during a "crunch"), to monitor progress, set policy, commit resources as needed, resolve any differences of opinion that could not be resolved by the Project Team, and approve/disapprove Project Team recommendations. It should not get involved in the day-to-day implementation effort; that is why the Project Team exists. The Steering Committee must also ensure that adequate contingency plans, internal controls, and documentation exist as the system is being designed and installed.

CONTROLLING

There can be no control without adequate plans, for one must control to a predefined goal. There can be no controls without proper organization, for there would be no person held responsible. Given, however, that plans and organization exist, control is absolutely mandatory. Without control, there is no feedback to inform management of deviations from plan to allow them to redirect the implementation efforts appropriately, or to measure the performance of the persons involved. Of the three management functions, controlling is the most difficult, and the one that is least well executed, in my experience. More implementation efforts fail from lack of adequate control than from the other two functions combined.

We have discussed earlier that the Project Manager, and Project Team, must control the project on a daily basis. They must monitor progress against each of the major requirements:

1. *Time*. To do this, each project must be subdivided into tasks so small that each of them takes one person no longer than two weeks. Each of these tasks needs to be identified on a PERT chart, staffed, and tracked. This avoids the surprise of learning, one week before scheduled conversion, that the project is six months late. Progress must be reviewed weekly.
2. *Budget*. The easiest way to monitor this is to use project control software. Expenditures must be reviewed weekly, in concert with progress and projected completion dates.
3. *Benefits*. These need to be followed also, for if they are not going to be achieved, the project should be considered for immediate discontinuation by the Project Team and Steering Committee.
4. *System Performance*. Same as Benefits. If the system will not perform as expected, implementation should be stopped unless reapproved by the Steering Committee.
5. Internal Controls, and Contingency Plans. In the euphoria of system development, nobody wants to think about such things. They are absolutely essential. Internal controls can, and do, highlight system deficiencies. After our new Order Entry system had been installed for a year, our Controller insisted on installing another simple internal control. It revealed that on a very few occasions, we were not invoicing our customers for goods shipped! Contingency plans are required, because the hardware will eventually fail. (Murphy was correct; ours failed during our monthly close.) We are still in business because we had developed contingency plans.

Let me reemphasize that the Project Manager and the Project Team need to continually keep the project boundaries in firm focus. I suspect that more projects have floundered and finally sunk from the mid-stream addition of features, enhancements, etc. than from any other single cause. Once the Functional Specification is approved, there should be no major changes without Steering Committee approval. Once the Detail Design is finalized, there should be few if any changes allowed.

If a package is being installed, requests for change should be segregated into three categories: a) Must Have Before Implementation, b) Should Have As Soon As Possible, and c) Nice To Have. There should be very, very few changes in category a); these are the ONLY changes that should be permitted before implementation of the standard package. Once the package is installed and running, over half the requests in categories b) and c) will disappear; they will no longer be necessary. Each change that is permitted to delay the installation of the package delays the benefits that will be derived from installation, and increases future maintenance problems.

The Steering Committee must measure progress against plan for all major dimensions outlined above, and ensure timely completion to specification. They must resist the overwhelming urge to modify, or enhance, unless the benefits are extremely attractive. They must be willing to scrap the project if the costs grow, as is usually the case, and the benefits shrink, as is also usually the case, to the point at which it is no longer financially attractive, as is fortunately the case only occasionally. Finally, they must ensure that old systems are left intact until the new system has proven that it really works. I visited a company some years ago that demonstrated the validity of this last point. They had destroyed the old system; the new one had not worked for two months. The people in the plant were playing cards.

CONCLUSIONS

Systems do not implement themselves; people implement them. To succeed, a systems implementation effort must be managed effectively, by applying standard management principles (Planning, Organizing, and Controlling) with the intent to minimize risk. This is accomplished by using a time-phased commitment approach that provides management three separate opportunities to review costs and benefits and schedules, and to discontinue the effort with only the minimum possible resources having been expended at each of those decision points.

It is imperative that we, as MIS professionals, cause systems to be implemented properly in our respective companies. Our companies cannot afford the disaster of systems implementation failure. We cannot afford the continued negative publicity, and the resultant scepticism concerning our professional competence. Or, to be more blunt, a manager is only as good as his ability to deliver on his promises; we have proved for 20 years that we still lack that ability. It is time for us to acquire it, or face the consequences.

BIBLIOGRAPHY

Bliss & Laughlin Industries. Corporate Data Processsing Standards Manual. Oakbrook, Illinois.

Edson, Norris W., "The Realities of Implementing MRP," 23rd Annual Conference Proceedings (1980), American Production and Inventory Control Society, Inc., Washington, D. C.

Jones, Gary D., "Pitfalls to Avoid in Implementing MRP," 21st Annual Conference Proceedings (1978), American Production and Inventory Control Society, Inc., Washington, D. C.

Lasden, Martin, "Turning Reluctant Users On To Change," Computer Decisions, January 1891, pages 92-100.

McFarlan, F. Warren, "Portfolio Approach to Information Systems," Harvard Business Review, September/October 1981, pages 142-150.

Olsen, Robert E., "MRP Implementation — Doing It The User Way," 21st Annual Conference Proceedings (1978), American Production and Inventory Control Society, Inc., Washington, D. C.

Orr, Kenneth T., "Systems Methodologies for the 80s," Infosystems, June 1981, pages 78-80.

Salmere, Mitchel B., "How to Improve a Management Information System," Infosystems, November 1981, page 90.

Wight, Oliver. The Executive's New Computer, Reston Publishing Company, Reston, VA, 1972

APPENDIX A1

The Functional Specifications can include any and all of the following topics:

- General Background
- Management Summary
- Problem Definition
- Present System Description
- Major System Objectives
- Proposed System Description
- Economic Justification
- Detailed Plan of Action
- Responsibilities
- Proposed Schedule

APPENDIX A2

Functional Specifications for a system can include the following topics:

- Major Logic Chart(s)
- System Narrative
- Design Notes and Concepts
- Proposed Input and Output Layouts
- Proposed Controls
- Anticipated Throughput Volumes
- Future Capabilities
- Environmental Constraints on Expansion Capabilities
- Hardware and Software Considerations
- Proposed Training Plan
- Cost Considerations and Assumptions
- Interface Considerations
- Audit Considerations
- Contingency Plan

APPENDIX A3

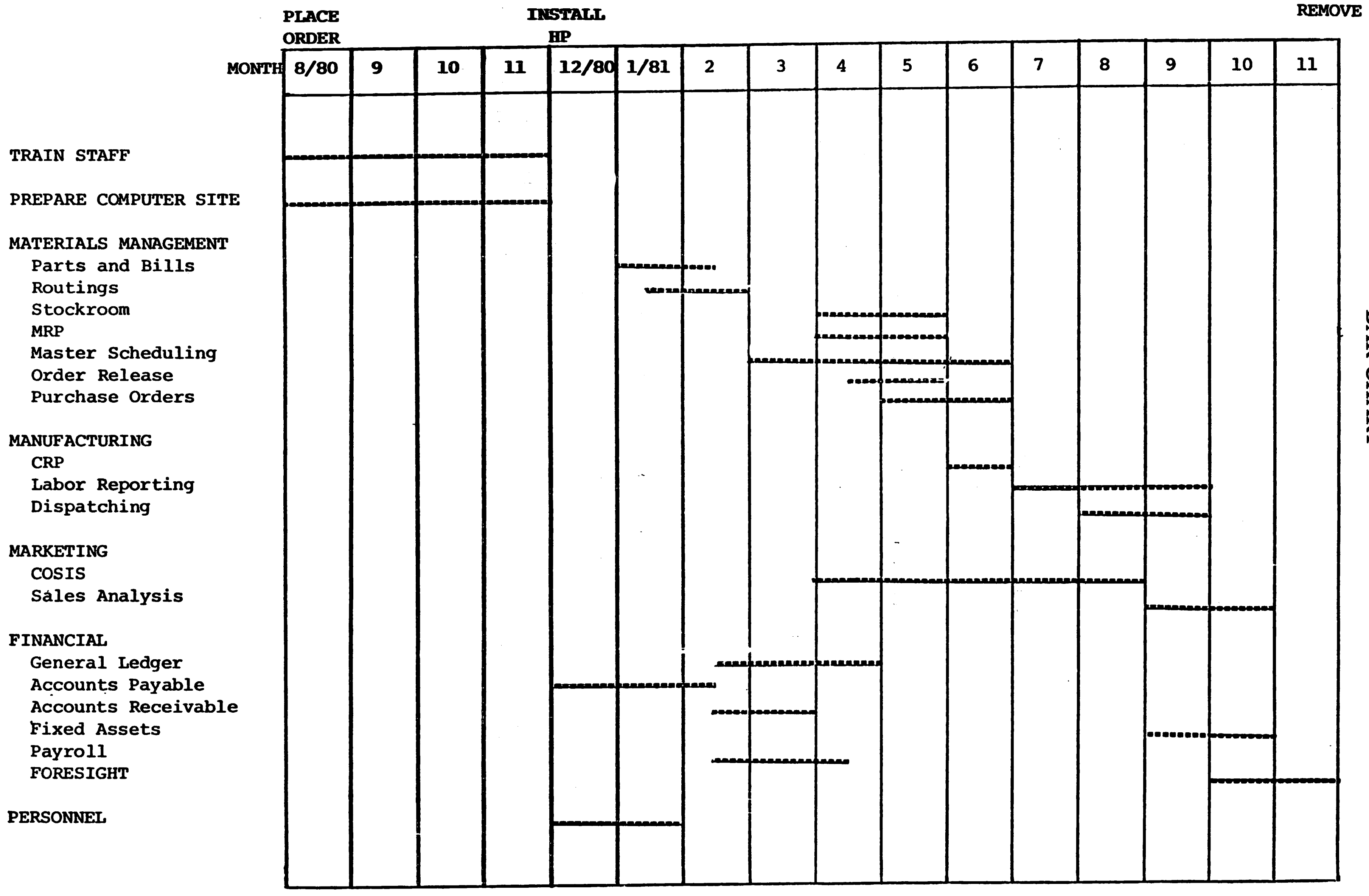
These items should be included in a Detail Design Specification; others may be added at your discretion:

- Detailed System Flowchart, defining input and output for each program
- Detailed Narrative for each section of the flowchart
- Program Run Sequences
- Audit Measures
- Quality Control Measures
- Internal Control Measures
- Security
- File and Data Conversion from the Present System
- Recovery Procedures
- Programming Conventions
- Test Specifications
- Test Standards
- Hardware/Software Environment
- Program Narratives
- Program Functions
- Program Specifications
- Program Logic
- Detailed Project Implementation Schedules

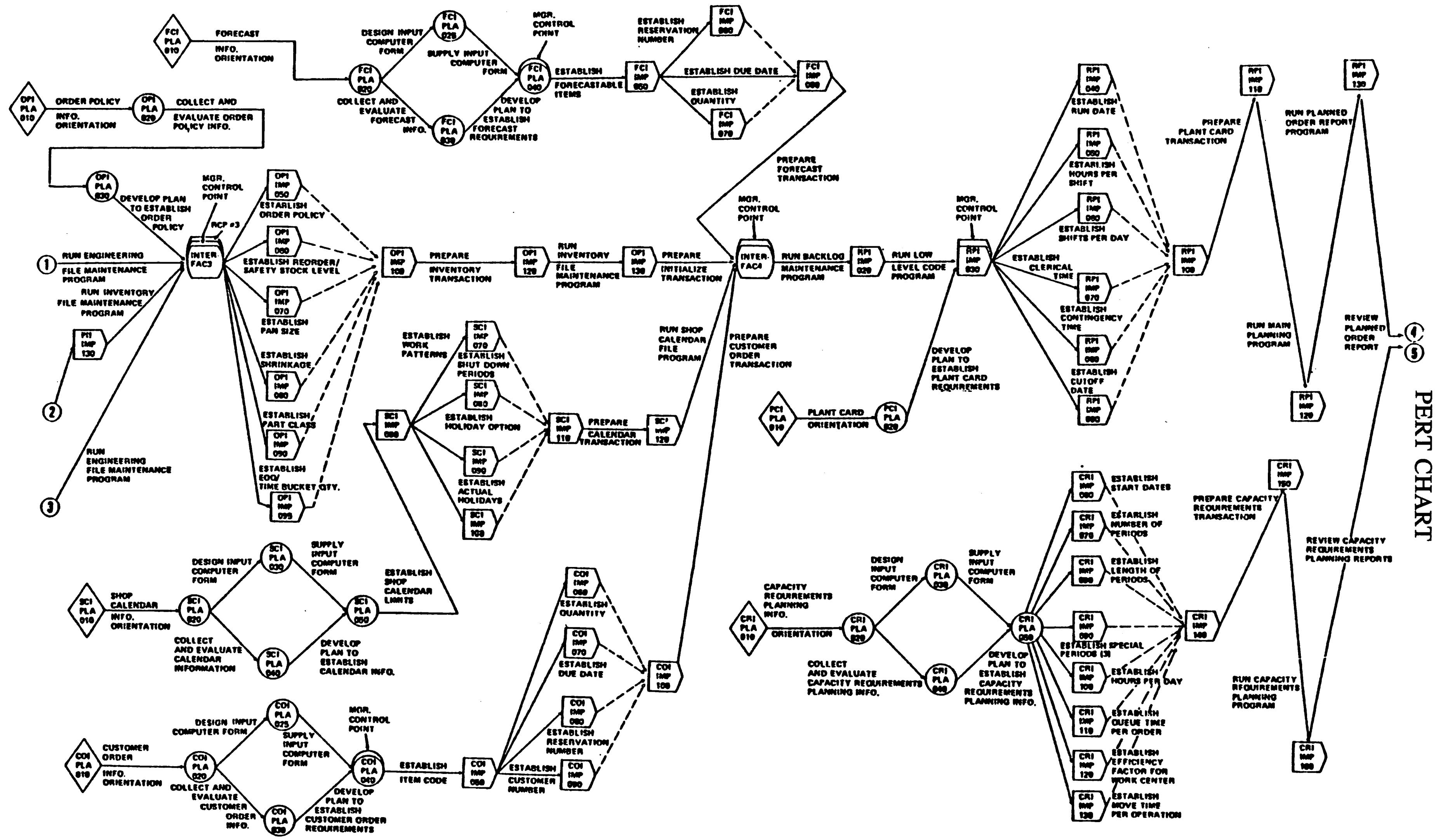
APPENDIX B
BAR CHART

SYSTEM IMPLEMENTATION MILESTONE CHART

(Assumes BLI approval not later than 8/1/80)



APPENDIX B
PERT CHART



FCI PLA 010
FORECAST INFO. ORIENTATION
DESIGN INPUT COMPUTER FORM
FCI PLA 020
FCI PLA 025
SUPPLY INPUT COMPUTER FORM
COLLECT AND EVALUATE FORECAST INFO.
FCI PLA 030
MGR. CONTROL POINT
FCI PLA 040
DEVELOP PLAN TO ESTABLISH FORECAST REQUIREMENTS
ESTABLISH FORECASTABLE ITEMS
FCI IMP 050
ESTABLISH RESERVATION NUMBER
FCI IMP 060
ESTABLISH DUE DATE
ESTABLISH QUANTITY
FCI IMP 070
FCI IMP 080
PREPARE FORECAST TRANSACTION
OPI PLA 010
ORDER POLICY INFO. ORIENTATION
OPI PLA 020
COLLECT AND EVALUATE ORDER POLICY INFO.
OPI PLA 030
DEVELOP PLAN TO ESTABLISH ORDER POLICY
MGR. CONTROL POINT
RCP #3
INTERFAC3
RUN ENGINEERING FILE MAINTENANCE PROGRAM
RUN INVENTORY FILE MAINTENANCE PROGRAM
PII IMP 130
RUN ENGINEERING FILE MAINTENANCE PROGRAM
OPI IMP 050
ESTABLISH ORDER POLICY
OPI IMP 060
ESTABLISH REORDER/SAFETY STOCK LEVEL
OPI IMP 070
ESTABLISH PAN SIZE
ESTABLISH SHRINKAGE
OPI IMP 080
ESTABLISH PART CLASS
OPI IMP 090
ESTABLISH EOQ/TIME BUCKET QTY.
OPI IMP 099
OPI IMP 100
PREPARE INVENTORY TRANSACTION
OPI IMP 120
RUN INVENTORY FILE MAINTENANCE PROGRAM
OPI IMP 130
PREPARE INITIALIZE TRANSACTION
MGR. CONTROL POINT
INTERFAC4
RUN BACKLOG MAINTENANCE PROGRAM
RPI IMP 020
RUN LOW LEVEL CODE PROGRAM
MGR. CONTROL POINT
RPI IMP 030
RPI IMP 040
ESTABLISH RUN DATE
RPI IMP 050
ESTABLISH HOURS PER SHIFT
RPI IMP 060
ESTABLISH SHIFTS PER DAY
ESTABLISH CLERICAL TIME
RPI IMP 070
ESTABLISH CONTINGENCY TIME
RPI IMP 080
ESTABLISH CUTOFF DATE
RPI IMP 090
RPI IMP 100
PREPARE PLANT CARD TRANSACTION
RPI IMP 110
RUN MAIN PLANNING PROGRAM
RPI IMP 120
RUN PLANNED ORDER REPORT PROGRAM
RPI IMP 130
REVIEW PLANNED ORDER REPORT
PCI PLA 010
PLANT CARD ORIENTATION
PCI PLA 020
DEVELOP PLAN TO ESTABLISH PLANT CARD REQUIREMENTS
ESTABLISH WORK PATTERNS
SCI IMP 070
ESTABLISH SHUT DOWN PERIODS
SCI IMP 080
ESTABLISH HOLIDAY OPTION
SCI IMP 090
ESTABLISH ACTUAL HOLIDAYS
SCI IMP 100
SCI IMP 060
SCI IMP 110
PREPARE CALENDAR TRANSACTION
SCI IMP 120
RUN SHOP CALENDAR FILE PROGRAM
PREPARE CUSTOMER ORDER TRANSACTION
SCI PLA 010
SHOP CALENDAR INFO. ORIENTATION
SCI PLA 020
DESIGN INPUT COMPUTER FORM
SCI PLA 030
SUPPLY INPUT COMPUTER FORM
COLLECT AND EVALUATE CALENDAR INFORMATION
SCI PLA 040
DEVELOP PLAN TO ESTABLISH CALENDAR INFO.
SCI PLA 050
ESTABLISH SHOP CALENDAR LIMITS
COI PLA 010
CUSTOMER ORDER INFO. ORIENTATION
COI PLA 020
DESIGN INPUT COMPUTER FORM
COI PLA 025
SUPPLY INPUT COMPUTER FORM
COLLECT AND EVALUATE CUSTOMER ORDER INFO.
COI PLA 030
MGR. CONTROL POINT
COI PLA 040
DEVELOP PLAN TO ESTABLISH CUSTOMER ORDER REQUIREMENTS
ESTABLISH ITEM CODE
COI IMP 050
COI IMP 060
ESTABLISH QUANTITY
COI IMP 070
ESTABLISH DUE DATE
COI IMP 080
ESTABLISH RESERVATION NUMBER
ESTABLISH CUSTOMER NUMBER
COI IMP 090
COI IMP 100
CRI PLA 010
CAPACITY REQUIREMENTS PLANNING INFO. ORIENTATION
CRI PLA 020
DESIGN INPUT COMPUTER FORM
CRI PLA 030
SUPPLY INPUT COMPUTER FORM
COLLECT AND EVALUATE CAPACITY REQUIREMENTS PLANNING INFO.
CRI PLA 040
CRI PLA 050
DEVELOP PLAN TO ESTABLISH CAPACITY REQUIREMENTS PLANNING INFO.
CRI IMP 060
ESTABLISH START DATES
CRI IMP 070
ESTABLISH NUMBER OF PERIODS
CRI IMP 080
ESTABLISH LENGTH OF PERIODS
CRI IMP 090
ESTABLISH SPECIAL PERIODS (3)
CRI IMP 100
ESTABLISH HOURS PER DAY
CRI IMP 110
ESTABLISH QUEUE TIME PER ORDER
CRI IMP 120
ESTABLISH EFFICIENCY FACTOR FOR WORK CENTER
CRI IMP 130
ESTABLISH MOVE TIME PER OPERATION
CRI IMP 140
PREPARE CAPACITY REQUIREMENTS TRANSACTION
CRI IMP 150
RUN CAPACITY REQUIREMENTS PLANNING PROGRAM
CRI IMP 160
REVIEW CAPACITY REQUIREMENTS PLANNING REPORTS
1
2
3
4
5

PANEL C

PANEL D