A GUIDE TO SYSTEMS DEVELOPMENT

JOHN M. GRILLOS
AMERICAN MANAGEMENT SYSTEMS, INC.

INTRODUCTION

Developing a large scale computer application is an expensive effort, requiring careful, planned coordination of many different activities to be successful. This paper outlines the overall approach and key activities essential to the orderly development of a large system.

Not all management problems require computer-based solutions; careful analysis can sometimes yield significant improvements simply through new policies or procedures. However, if the problem is regular and recurring, involves large amounts of data, or requires special monitoring and control, automation should be considered.

This paper segments the development of a new system into three major phases:

-- System Concept (Phase I)

-- System Design (Phase II)

-- System Implementation (Phase III)

These phases reflect the idea that development is a learning and refining process, in which a few good ideas are expanded and detailed as more information becomes available to the point where millions of orderly pulses racing around in a computer perform the few functions intended when the development process began.

Good planning is essential to the whole process. Each project requires a carefully designed plan, giving consideration to task dependencies, available resources, non-system constraints, and management requirements. The guidelines presented here are a starting point for a task plan and a measure of task completion and project status. The milestones defined in this paper are usually accomplished regardless of the size of the system. However, smaller systems quite often allow collapsing of two phases or subphases of development. These are individual considerations which go into developing a plan.

Good documentation standards are also critical to the development process. Each project task must be documented. Documentation reduces potential confusion in verbal communication, broadens exposure to and makes explicit decisions, and allows measurement of successful completion of most non-programming tasks. This paper gives an indication of the type of information to document. Other areas in which standards are useful include: Naming conventions for data elements, segments, files, data bases, data glossary, and so on; Programming techniques for structure, frequency of comments, and naming of COBOL paragraphs; Formatting for reports, forms layouts and so on; and Environmental considerations such as operating standards, turnover standards, programming languages, use of data base, and so on.

A procedure for review and revision of all task documentation by both technical project management and concerned users should be in place. This is supplemented by substantive progress briefings periodically delivered to management. Disagreements and misunderstandings which persist through system development because of poor review procedures are very expensive to resolve in an operational system.

Beyond the process and ideas above are the following considerations which are also keys to the total success of a system development.

1. Personnel Assignments -- Beyond quality of people assigned is their orientation and training. Generally, business analysis skills are most useful in doing concept and general design work, and of course technically qualified professionals dominate staffing of detail design and implementation phases -- with business types once again involved in training and installation. However, involvement of technical talent in early project phases can head off problems caused by bad technical assumptions upon which important application functions may depend. Involvement of business-oriented personnel in late stages of development provides project continuity and, thereby, insures that user requirements stated during General Design are faithfully adhered to in subsequent project phases.

2. User Involvement -- System projects have little hope of achieving intended benefits without adequate user involvement. Users must be the primary definers of system functionality, and must be involved in critical review processes that occur at points throughout the development cycle. These requirements mean two things -- users must be actively interested in the project so that they are willing to spend the time to stay involved, and project planning must deal explicitly with user level of effort and timing considerations so that users can schedule project involvement into their routines.

3. Budgeting Process -- With good planning it is reasonable to expect reliable cost estimates for one phase beyond the current phase (e.g., reliable Detail Design costs can be prepared at the end of General Design). However, budgeting for an entire development cycle in advance or budgeting two phases into the future cannot be done with much reliability. Vagaries in staff productivity or costs, need to do tasks not foreseen in plans, personnel turnover, and a variety of other unforeseen problems seem to always overtake project cost estimates. Since full project budgets cannot be made accurately in advance, two things should be done -- budget estimates should be revised after each phase for management go/no-go decision making, and projects should not be hamstrung by attempted enforcement of premature and inadequate budgets. Such enforcement can result in no system or a system which does not meet important project objectives.

The following discussion breaks each development phase into several types of tasks. Those tasks that would be repeated during the same phase for each of several subsystems comprising a larger overall system are indicated by an asterisk (*),

PHASE I -- SYSTEM CONCEPT

The purpose of the System Concept is to analyze a problem, define its cause, and create an economic solution, which may utilize the computer. The emphasis is on economic feasibility, but technical considerations are recognized in order to analyze the cost of any solution. However, these considerations are only examined in enough detail to derive rough measures. This phase is the first pass at a complete design.

A. Define Scope and Effort

The first task is to make estimates of the level of effort to be expended in examining the problem. A plan must be developed scheduling the tasks described in the remainder of this section.

B. Review of Current System

The current system should be examined to provide a base line against which the value of a new system can be measured. The current system includes all manual and computer systems considered for change. Some necessary information to be documented includes a description of processing, inputs, outputs, processing volumes, personnel utilization, computer utilization, and costs.

User review of all documentation should be done. Documentation should be reviewed quickly, yet in detail. The outputs and functions defined here provide the base line for measurement and minimum requirments of any new system.

C. Determine Analysis Areas

Based on the above review, several key areas usually stand out as being points where additional analysis is critical for system success. These "leverage points" may have any of the following characteristics to distinguish them from other potential concerns:

1. High Volume Processes -- where small improvement can yield high return.
2. Simple Clerical Tasks -- are usually easy to mechanize.
3. Timelines and Quality of Input Data -- the cost of processing bad data for correction is usually excessive.
4. Cash Management Benefit -- earlier billing and later payment have measurable savings.
5. Timelines of Reports -- untimely reports usually force a secondary system to be created for control.
6. Reduction of Codes -- several codes used to define the same characteristic are very frustrating to users.
7. Management Interest -- any point where a concern has been specially expressed should be analyzed.

D. Analysis

Analysis is the key step. The areas defined in the previous task are investigated in detail and alternatives for providing the required service in an efficient manner are considered. Analysis content will vary widely from project to project, and the process must be documented. The goal is to define a feasible solution which is economically superior to the present system and other alternatives. Each alternative must be described in functional terms, stressing the comparison with the present system and other alternatives. The cost of operating the system once it is in place and operating must be determined. The level of detail should not be too great; if the economic feasibility of an alternative is so marginal that it hinges on a very detailed analysis, the alternative should be avoided. All benefits should be quantified if possible. Personnel reductions, cash flow, and inventory reductions are obvious examples.

Reports can be evaluated in terms of the action that could be taken if the information were available. However, it is usually best not to attempt to quantify their value. Any future benefit possibilities should be assigned an expected value. Cost avoidance should also be quantified and stressed.

E. New System Concept

The results of the analysis areas are integrated into a new total system concept which also incorporates the definition and analysis of common system components which serve the entire system, such as report generators or data bases which were not explicitly considered before. Other elements of the concept are a system flowchart, narrative description which emphasizes the flow of data and processing of major exceptions, a list of features, a list of remaining issues and criteria for resolving them, and a system operating cost/benefit summary.

F. Phase II Task List

The next task is to create a task list for the design of the system. Depending upon the size of the project, the task list may include General Design or both General and Detail Design. The task list includes the amount of resources required for completion.

G. System Development Cost and Schedule

Development cost for the second phase is derived from the task list by assigning a cost to the resources required. In order to estimate the cost of later phases, an estimating procedure based on the number of major files, programs, reports and forms can be used. Table 1 shows some planning factors which may be adjusted by experience to the personnel actually involved. The resources required for Phases IIIB and IIIC can only be estimated by personnel familiar with the user's environment.

Since the system is very sketchy at this time, the number of files, programs, reports and forms may not be accurately estimated. To correlate the costs developed this way, Table 2 can be used as a rough measure. If results generated from Table 1 differ greatly, well-defined reasons should be required.

TABLE 1

Development Cost Estimation

(Man-Days)

	Major Files	Major Programs	Minor Programs	Reports and Forms	Project[1] Management
General Design	Task List Expansion				20%
Detailed Design	15	10	5	3	20%
Programming and Unit Test		35	12		20%

To correlate the estimate generated from the "bottom up" approach, the following guidelines can be used as indicators of distribution of cost between phases.

TABLE 2

Phase I	5%
Phase IIA	10%
Phase IIB	20%
Phase IIIA	35%
Phase IIIB	15%
Phase IIIC	15%

[1] Project Management is calculated as a percent of man-days on other tasks. As the size of the project grows above 20-25 personnel assigned, extra management above this may also be required.

Schedule should be defined at this time. The development sequence of each subsystem including precedences and implementation priorities can be illustrated by a staging chart. A required implementation date, a limit on the maximum personnel assigned to a project, or a known team are possible constraints to the schedule. The goal is a rough schedule for initial planning purposes.

Finally, return on investment or other financial viability measures should be computed using quantified system benefits, operating costs and development cost schedule.

H. Management Presentation

As stated earlier, user involvement and management review is critical. The review at the end of the system concept is the key to beginning the actual design project.

PHASE II -- SYSTEM DESIGN

The goal of the design phase is to devise the most effective and efficient means to provide the services which were earlier defined as economically desirable. Where the System Concept analyzed economic feasibility within wide limits of technical capacity, the System Design phase analyzes technical feasibility of alternative methods to provide the desirable features. If the project is large, the design phase is broken into two subphases: General and Detailed Design.

A. General Design (Phase IIA)

The General Design phase is user oriented. All user issues should be resolved. Manual interfaces and processes, code types to be used and reporting needs are defined. The end product of the General Design is a specification of user requirements and interfaces, and a defined solution to the major technical problems of the system. The technical feasibility of the design must be thoroughly analyzed and alternative technical approaches devised and evaluated. The following tasks are usually required:

1. Planning

The task list from System Concept must be expanded into a working plan for this phase. Personnel must be matched to tasks, tasks grouped and assigned for effective continuity, and task interdependencies considered. Since integration of the subsystems is a primary goal of General Design, significant time must be reserved for analysts to review each other's work. Review procedures for both users and technical analysts should be precisely defined here, as well as procedures for resolving stalemates.

2. Issue Analysis

The issues to be resolved were defined in the System Concept phase. This analysis is quantified as much as possible, though these issues are generally harder to assign dollar values. A mechanism for resolving issues must be defined. Usually, a specific person is designated to approve issue resolutions. It is also critical to impose and document deadlines for issue resolution.

*3. Define Inputs and Outputs

The problem of which functions must be defined first (input, output or process) continually arises. It is often a restatement of the "chicken or the egg" riddle. The key here is to pick a sequence, and proceed with efficient timely analysis. In general though, a firm grasp of user reporting needs serves as a good starting point.

Codes to be used in the system must be defined as the design progresses. This task is completely overlapped with file design, and is also sometimes a major issue. There must be some explicit addressing of the use and purpose of each major code.

Report definitions should include: narrative descriptions, contents, purpose, users, frequency, and media. Files (both computer and manual) must be adequately described, including:

- Type of file.
    - System Master Files. Most of the subsystems use these files, though they are normally built and maintained in a separate subsystem. Designers of master file subsystems are often key personnel who insure system integration and finalize code definition. Master file definition is especially difficult to schedule since master file requirements may change several times during the design stage because of issues arising in processing subsystem design or technical problems which surface during the design.

  - Subsystem Master Files. Used totally within a single subsystem; and contain relatively stable information to be updated and maintained by the subsystem.

  - Input Files. Information from other subsystems is defined by the subsystem requiring the information, except, of course, at the initial entry of a source document.

- Description of files.

  - Description of Use. Narrative format.

  - Data Elements. Description, rough size estimate and frequency.

  - Records. Grouping of elements.

  - Processing Intent. A description of what the file is used for and how it is to be processed (e.g., direct or sequential).

  - Organization. Definition of how the file may be organized, e.g., IMAGE, KSAM.

Input documents must be described. The level of detail varies with importance to the system, and the system operating environment. Critical input documents should be roughed out and approved by the user. Occasionally, field testing is required, but usually critical documents should not be finalized in this phase.

Communciations requirements should be defined. Aspects to be considerpd and documented include rough estimates of line loading, location and types of terminals, transmission rates, recovery feasibility, fall back procedures, and alternative approaches.

*4. Computer Processing

Computer processing is described in terms of subsystem functions, not programs. The division of these functions into programs does not take place until Detail Design.

Subsystem flowchart definitions describe process functions connecting inputs and outputs. The functions are simply logical groupings of data manipulation, and definition of these is the key talent of a system designer.

Function definitions are detailed narrative descriptions of what each function will do. This does not mean a decision block level of flowchart or description. It must be in terms understandable to a non-technical system user.

Computer resource estimates based on the frequency of processing cycle and a profile of the inputs should be made. If technical feasibility of any alternative appears uncertain as a result of this analysis, further refinement by the use of simulation, emulation or more detailed analysis may be required. Establishment of technical feasibility is a requirement of this stage.

*5. Manual Procedures

Manual procedures related to system function should be documented. A processing flowchart should be prepared depicting operations to be performed and organizations which will handle inputs and outputs. Detailed narrative descriptions should be used to describe coding information on forms, balancing needs, error correction, data entry and verification or validation. Personnel requirements over daily and monthly cycles, including profiles of activity over time (peaks and averages) should be defined.

6. Environmental Conditions (System Configuration)

The tasks in this area are concerned with technical support for the development and operation of the system.

- Software Environment -- or system support utilities as well as more general installation support is analyzed here. Alternatives must be considered when differences in cost or maintainability exist. Objects of analysis may include: data base approach, use of a report generator, inquiry facilities, use of a test data generator, and programming languages.

- Training -- of programmers and analysts for later development phases should be planned and begun, if possible.

- Hardware Environment Analysis -- should be performed at the end of the General Design since it requires evaluation of all subsystems. If alternative hardware is under consideration, system resources or needs must be analyzed, including on-line storage, CPU utilization, (total hours required per processing cycle), I/O elapsed time, order lead time, and communications support needs.

7. System Flowchart

This is a key ouput from the phase. It should be understandable to both user and computer personnel, and provide an interface for discussion purposes. Two levels should be produced. A summary level chart reworks the Phase I flowchart, with the same features, issues, descriptions, etc. An integrated detail chart links subsystem flowcharts and is used to check for integration. All detail subsystem charts are included on this consolidated version.

8. Cost Benefit

This is reworking of operating cost and benefits in light of new detail to insure continued economic feasibility.

9. Phase IIB Task List

A detailed task list and resource estimate is now required. This task list may still be in terms of process functions rather than detail components, depending upon staffing uncertainties, start date for next phase, etc. If so, the list will be expanded early in Detailed Design, as discussed later.

10. Priorities for Development and Implementation

This is a refinement of the initial discussions with user groups, and must be reviewed with them. Precedences may be clearer and costs of partial implementation can now be considered. Priorities may not indicate a rigid scheduling procedure, due to alternative distributions of project resources (e.g., if one subsystem can only use four analysts and six are available, then a second subsystem can be worked on). Priorities act as guidelines in assigning scarce resources to the project. Some factors to consider are extra bridge (or conversion) programs, revision to high priority programs when the full system is in place, early realization of benefits and progress visibility.

11. Development Cost and Schedule

The development cost can now be re-evaluated with the detailed Phase IIB task list. A further refinement of Phase III costs is also possible since a firmer estimate of computer programs is possible. A rough schedule must also be defined utilizing the priorities and precedences known.

12. Management Review

The management review process has only to cover changes in scope, technique or benefits from the Phase I presentation. A progress report may be all that is required if changes are minimal. However, technical feasibility must be firmly established at this stage and should not be in doubt upon management review.

B. Detailed Design (Phase IIB)

The Detailed Design phase of a systems development project develops system specifications in sufficient detail to accomplish the programming and implementation tasks. If the system is small, then separating the design into two sub-phases may not be necessary. In any case the outputs should be the same. If the implementation plan places development priorities on the various subsystems, the higher priority subsystems should have their designs completed before the lower priority subsystems.

In this phase, process functions are grouped into programs, and detailed specifications are developed for these programs. Rough report formats are developed into report layout specifications, and all reports, including activity lists, are specified. Input forms are finalized. File contents are refined to detailed file specifications, and layouts are prepared for all files. Manual procedures are further detailed to the level of detailed job specifications.

Non-application activites of this phase include making final arragements to insure the availability of hardware and software (including communications facilities) to provide a testing and implemention environment. Also, a series of final volumes are prepared which serve as the system documentation for implementation and later maintenance.

The key aspect of Phase IIB then is a further refinement of design and planning work done up to this point.

The level of detail required for a detailed program specification is determined by two things:

-- The level of qualification of the programmers who will have to program from these specifications; the more senior, the less detail required.

-- The staff continuity between Phase II and III. That is, if the same people who are designing the subsystem will also be programming the subsystem there is a reduced need for detail.

The following sections discuss the items to be accomplished during Detailed Design.

1. Detailed Project Plan

A detailed project plan must be prepared to control the Detailed Design Phase. This plan is based on the task list prepared at the end of Phase IIA, and reflects the priorities of subsystem development.

2. Resolve Remaining Issues

There will surely be some issues remaining from Phase IIA. These should receive the highest priority for resolution before proceeding, and be settled very rapidly.

3. Codes

It is necessary to define what new codes will represent. Consideration may have to be given to alternative coding types (structured codes and random coding values), specifying the length and composition of each alternative. Give consideration to the integration of new codes; that is, if a similar code is used elsewhere in the same business, try to maintain a consistency of the structure, and assigned values.

4. Detail Data Base Design

Although this section is primarily devoted to considerations of a data base system, some of the concepts are useful and should be considered even when not in a "formal" data base system:

- Produce a glossary of terms for data elements for the project. This glossary will be constantly maintained throughout the project (and possibly throughout the use of the facility). It should include the element name, a description, its "picture", the source of the element, (eventually) which files it is in, and where its (subsystem) maintenance responsibility lies.

- For large projects, data base design is a further (but not final) refinement of the concepts and data needs studied in Phase IIA, and includes developing an overall data base design, examining data relationships both inside and outside the project, and developing initial data base size estimates. Consideration should be given to redundancy and efficiency tradeoffs and integration with other system DB requirements. Size estimates should attempt to deal with overhead storage for system flags and pointers.

- The final data base design is done later in Phase IIB, and should include finalized physical and logical layouts of the data base, access methods, estimates of the frequency and timings for backing-up, recovery, and reorganizing the data base, and an outline of recovery procedures.

- If necessary, select SYSGEN options to optimize the performance of the data base facility for your project or system. TOTAL has relatively few SYSGEN options, whereas IMS has many.

*5. Detailed Subsystem Flowchart

The starting point for all detailed design work is a detailed subsystem flowchart, showing all inputs, programs, files, reports and other outputs of the subsystem, and how they connect. There must be firm agreement between technical staff and the users regarding this subsystem flowchart before proceeding with Detailed Design.

*6. Inputs/Outputs

For each subsystem, the inputs and outputs must be fully specified in detail -- this includes reports, files (data bases), and input (source) documents. Each of these are treated below:

- Reports must be specified in detail. Such things as a detailed report layout (with descriptions of the contents of each field on the report), a general description of the contents and purpose of each report, an estimate of the volume and frequency of generation, output medium, and special code values (e.g., error codes) should be indicated in detail.

- File specifications (layouts) consist of at least the details of the data elements and their groupings (e.g., COBOL-like layout or data base layout). The data element lengths, type relative positions, and frequency (of specific groupings or record types) should be given, along with record lengths and blocking factors.

- Input (source) document formats should be finalized at this point. Besides form layouts, keying formats, revised volume and timing estimates, and a chart which cross-checks the data elements with the source documents to insure that all data required is being captured should be prepared.

- Communications steps include revising volume estimates, and other data, determining detailed back-up and recovery procedures, and, if there are dedicated facilities, detailing the entire communications network (lines, their locations, line type, conditioning, maximum transmission rates, modem requirements, multiplexing facilities, and any other data communciations control equipment requirements). A detailed cost analysis should be performed on this planned network, if not already done.

*7. Computer Processing

The initial task plan for Phase IIB which was prepared at the end of Phase IIA was process function oriented. However, the system is really program oriented. The relationships between these process functions and the programs to be specified have been given in #5 above. Now, detailed program specifications must be prepared, including:

- The program name, program identification, subsystem, inputs, and outputs.
- A brief description of the purpose of each program.
- Frequency of running and the estimated "stand-alone" run time.
- Recovery in the event of an external malfunction.
- A detailed program specification and corresponding detailed description should be prepared. As noted previously, the level of detail is primarily determined by the skills of intended implementors. Flowcharts will serve as the principal documentation of the programs for maintenance purpose. Therefore, accuracy is vital. A written narrative should be keyed to each box on each flowchart to elaborate on the meaning of that step.

*8. Manual Procedures

The need to have clear, workable manual procedures at the design stage cannot be overemphasized. General ideas of the manual procedures have been specified during Phase IIA, but these must be reviewed and detailed at this time. These procedures must be thoroughly reviewed and approved by the users. If possible, field tests of these procedures should be accomplished before their finalization.

9. Environmental Conditions

Technical support and planning must be provided in the software and hardware areas to insure that the necessary facilities are available for implementation of the system.

- <u>Software</u> support must be provided as indicated below:
  - Software decisions made in Phase IIA (e.g., data base management systems, reporting packages, etc.) must be reviewed and finalized. If they are no longer adequate, specific alternatives must be analyzed and recommendations made.
  - The software to support testing and implementation must be purchased and made available. The detailed implementation plan should be a guide as to when the various pieces of software must be available.

  - Basic technical support must be provided to the implementation, e.g., test data generators, standard error handling routines, program generators, data dictionaries, or special purpose routines. Because these utilities need to be available prior to programming, their development or installation should be started as soon as possible to insure that they are available when needed.

  - Test plans for the system software should be evolved and executed.

  - A detailed training program should be developed for the programmers, which would include familiarization not only with the new software systems (e.g., data base management systems), but also with the use of the software support utilities being developed.

- Hardware support must also be provided as follows:

  - The system resource requirements must be redone with the availability of the detailed design, and the CPU utilization, channel requirements, disks, tapes, terminals, and communications needs must be detailed. An order or modification of previous orders must be made. The delivery of hardware should be in phase with the needs for a testing, then an implementation environment.

  - Detailed plans for the physical facilities for the hardware (e.g., electricity, air conditioning, floor space, false flooring) must be made.

10. System Flowchart

Final subsystem documentation must be tied together to provide for easy availability for programming and maintenance. The items to be prepared are:

- A detail flowchart at the individual program level must be prepared for the entire system. This should be a combination of the flowcharts for the individual subsystem as done in #5 above. This flowchart should be backed up by a descriptive narrative, and supporting tables of programs and files. This flowchart is also very useful in ascertaining whether all subsystems interfaces are covered.

Final subsystem volumes (one per subsystem) should be prepared at the end of Phase IIB. These will serve as the final documentation for the subsystem, and should be a compendium of all files, programs, reports, and procedures previously prepared. Any later changes made during programming and conversion should be reflected in these volumes so that they can also be used to support maintenance.

11. System Implementation Project Plan Task List

A detailed task list must be prepared to cover the programming, systems testing, and conversion and training phases of the system development. The items to be covered in this plan are acceptance and performance criteria, coding and unit testing tasks, a detailed plan for program and system testing, and implementation phasing.

12. Management Review

A system overview presentation should be made to higher-level management and the users at the end of Phase IIB to gain specific approval to proceed to Phase III.

PHASE III- SYSTEM IMPLEMENTATION

Phases I and II accomplish the key tasks of definition and detailed specification of system scope, functions, and components. Phase III proceeds with these specifications to develop an integrated system of computer programs and manual procedures which is ready for installation in a production environment.

System Implementation consists of programming and unit testing, system testing, and conversion and training.

A. Phase IIIA -- Programming and Unit Testing

During this phase, detailed program specifications are analyzed, and individual programs are coded, compiled and tested.

1. Revise the Project Plan and Specify Detail Personnel Allocations and Work Schedules

2. Familiarize Programming Personnel with the Project

To ensure smooth project continuity, it is important to instruct programmers on system functions and components before they begin detailed coding. This is done by having them review detail specifications including relevant charts, file layouts and functional narratives, and then having programmers attend any formal training sessions planned for in Phase IIB (e.g., data base training courses).

*3. Program Coding

Detailed program specifications must be interpreted and translated into computer programs.

- Review detail program specifications.
- Where necessary, elaborate on program specifications in order to transform them into the working documents needed for program coding. At this point, it is useful to analyze input files, output files, processes, detailed calculations, and error conditions for correctness, consistency and potential use in program testing.
- Produce and compile program source code.
- Develop a plan for testing individual programs.

*4. Unit Testing

Individual computer programs are tested to insure that they meet program specifications.

- Install and checkout unit test software. This software, including utilities and aids to facilitate program testing, was defined and specified during system design.
- Generate unit test data. Files needed for program testing can be produced by both programmers, who generate files for a single program and a central technical support staff which creates system test master files used by groups of programs.
- Test programs and revise where necessary.
- Evaluate initial test volumes and timings in relation to the estimates developed during system design.
- Review unit test results and certify programs as ready for system testing.

B. Phase IIIB -- System Testing

Phase IIIA produces individual working computer programs which meet detail program specifications. System testing ensures that these programs and manual procedures operate together in the integrated manner necessary to accomplish system functions.

*1. Develop system acceptance criteria and test plan. The test plan normally covers such areas as input data to be used, expected outputs, interaction between computer system and manual procedures, system acceptance criteria, test plans for logical group programs, and a plan for phased, integrated system test.

2. Install system test software. These programs including utilities and aids to facilitate system testing were defined and specified during system design.

*3. Generate necessary input, master and intermediate system test data files.

*4. Conduct system test and revise programs where necessary.

- Analyze known inputs, resultant outputs, and the interaction between computer programs and manual procedures.
- Compare system test volumes and timings with benchmarks developed during system design and unit testing.
- Update system and programming documentation.
- Review system test in relation to system acceptance criteria, and obtain user approval for final implementation.

C. Phase IIIC -- Conversion and Training

1. Conversion

- Complete systems, operations and user documentation.
- Convert required master and systems files.
- Conduct pilot test as the first step of the phased system implementation plan. Steps include completing training of user personnel, installing any necessary pilot hardware of software, running the pilot test as a parallel run against an existing system (if applicable), and evaluating the pilot test in light of systems acceptance criteria. If necessary, revise programs, procedures, documentation and manuals based on results of the pilot run.

- Obtain system acceptance and begin full scale system implementation.

- Final implementation and signoff.

2. Training

- Develop a plan for training users, operations personnel and systems maintenance personnel.

- Conduct training sessions for users, operations personnel and systems maintenance personnel.