

2-1-1978

Computerizing Student Financial Aid

Richard H. Wedemeyer

Follow this and additional works at: <https://ir.library.louisville.edu/jsfa>

Recommended Citation

Wedemeyer, Richard H. (1978) "Computerizing Student Financial Aid," *Journal of Student Financial Aid*: Vol. 8 : Iss. 1 , Article 4.
Available at: <https://ir.library.louisville.edu/jsfa/vol8/iss1/4>

This Issue Article is brought to you for free and open access by ThinkIR: The University of Louisville's Institutional Repository. It has been accepted for inclusion in *Journal of Student Financial Aid* by an authorized administrator of ThinkIR: The University of Louisville's Institutional Repository. For more information, please contact thinkir@louisville.edu.

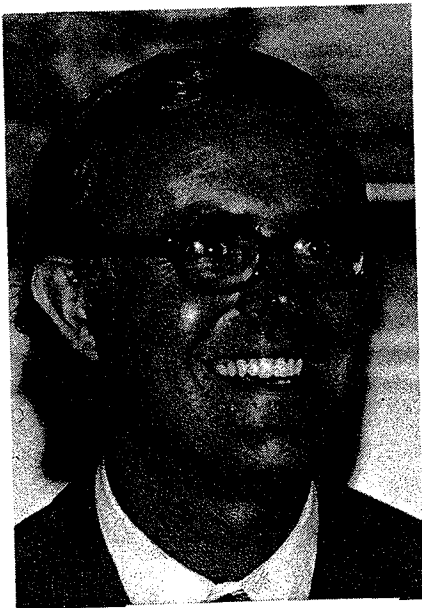
COMPUTERIZING STUDENT FINANCIAL AID

By Richard H. Wedemeyer

Student financial aid directors have found it increasingly difficult to manage their offices with the same resources they had five years ago. Additional programs, more aid applicants, and more stringent regulations have all contributed to the fact that more staff, a larger operating budget and possibly more space are often needed. A valid management question centers upon how efficiently present resources are being used. This search for efficiency has led many a financial aid director down the path of computerization.

The computerization of the financial aid process is often started and implemented without the necessary planning, organization or involvement needed to assure its success. Naive notions about data processing include the following: (1) system development will take only a few months; (2) all that must be done is to "push the button" and packaged awards will emerge from the computer; (3) activities will be simplified with a computer; and (4) no additional staff will be needed because this computer system will "do everything." These unrealistic expectations have caused many well-intentioned computer projects to fall miserably short of their original objectives. As a result, large amounts of staff time and resources are often spent in a highly unproductive manner. This paper seeks to identify some of the problems encountered in computerizing student financial aid and to suggest basic managerial concepts to insure a system's smooth development.

When computer projects fail, one can often trace the failure to a lack of realism about computer development. Much of this naivety exists because many financial aid officers lack data processing experience. As a result, expectations are often set too high and do not adequately reflect what can be accomplished realistically.



Dr. Wedemeyer is currently at Rutgers University as the Director of Financial Aid, Operations and Research. He received his Ph.D. from Northwestern University (Evanston, Illinois) and has previously worked for the Illinois State Scholarship Commission, Elmhurst College, and General Electric.

For example, the computer can be programmed to calculate the Total Family Contribution (TFC). On the surface this seems like a simple task. Provide a programmer with a copy of the College Scholarship Service (CSS) or American College Testing Program (ACT) computation manual and it should be programmed in two or three weeks. In fact, questions like these need to be answered: How will the computer be given the data from the Financial Aid Form (FAF) or Family Financial Statement (FFS)? What assumptions will be made if data are missing? What is the minimum amount of data needed? What kind of output is required for the financial aid operation?

In developing realistic expectations, it is desirable to obtain an estimate of the amount of time the systems analyst believes it will take to complete the project. One is better off over-estimating the time so that the target dates can be met for implementation. A realistic commitment of time on the part of the financial aid officer is required to insure success.

Since computerization is conceived of as a project which must be conducted by the data processing department, financial aid is often not totally involved in specifying what is needed. On the surface, it would seem easy to determine what is needed. For example, most aid officers want the computer to print an award letter. Many questions need to be answered before the level of detail is sufficient to start programming. These questions include: When is the award letter to be printed? What information should be included on the award letter? Where does this information originate? How can special messages be added to a student's award? How can you distinguish a revised from an original award letter? How will the student indicate acceptance or partial acceptance of the award? Does the computer retain the date when the award letter was printed? How many days will elapse from the award decision to the date the student receives the award letter? What staff resources will be needed to code, batch, keypunch and verify data?

For the most part, the answer to what is needed can be found in the current operational procedures of the aid office. Unfortunately, with the complexities of financial aid a thorough analysis of existing procedures is often obscured by federal or state regulations.

Often, there is a failure to ask the question of what should or should not be computerized. Below may be found some financial aid functions which can be computerized depending upon the environment. But should they be computerized now or later? Or with a particular office's operational environment, should they be computerized at all? These are the typical functions:

1. The automatic input of data from the following sources
 - a) CSS or ACT computer tapes or cards
 - b) Admission data
 - c) Registration data
 - d) Bursar's data
2. Automatic follow-up letters for more information
3. Folder labels
4. Calculation of TFC for documents not sent to CSS or ACT
5. Determination of budget

6. Determination of award eligibility
7. Automatic packaging of awards
8. Award letter
9. Tracking required forms
10. Cancellation notices
11. Automatic notification to the Business Office and possible credit to the general ledger
12. Check calculation and printing
13. Preparation of Federal Fiscal Operations Report and Tripartite Application for Student Aid Funds
14. Statistical retrieval
15. Address labels or lists of certain categories of students
16. Simulation capability for legislative advocacy
17. Award roster
18. Financial statement roster
19. Need roster (order of needy students)
20. Spending Summary

Even when an aid officer has described what is needed and what should be computerized, specificity is often lacking. The statements are so broad and vague that a great deal of misinterpretation results. This misinterpretation is often not discovered until the program is ready for implementation. The computer department is then blamed for failure when it was actually caused by the aid officer's lack of specificity.

For example, the statement that folder labels are needed is not specific and further detail should be added. Some questions that should be answered by the aid officer in this label example are as follows:

1. When should folder labels be produced? What triggers them?
2. What information should be included on the label?
 - a) student name?
 - b) social security number?
 - c) school code?
 - d) citizenship?
 - e) state of permanent residence?
 - f) address?
 - g) date FAF or FFS is received?
 - h) class year?
3. In what sequence should the labels be printed?
4. What are the procedures for handling the labels?

The constantly changing federal and state regulations contribute to the difficulty of computerizing financial aid. Changes in regulations is a basic axiom in financial aid. Once a computer program is written, it will perform the same steps and sequence of steps each time it is run. Modifying the data processing programs each year as regulations change wastes time. Too often, financial aid computer programs are not designed to be flexible. If new federal and state programs are added, many financial aid systems need to be rewritten because flexibility was not built into the original design.

A degree of flexibility can be incorporated by using a table of values rather than programming the values into the logic. In this way, the values of the tables (parameters) can be changed without having to rewrite the program. For example, budget values for various categories of students (i.e., resident/commuter, in-state/out-state etc.) can be incorporated in the table. Then if the budgets change, the table can be altered without reprogramming.

Awarding financial aid depends upon accurate and timely information from many departments within the college or university. Some of these departments include the business office, housing, meal services, admissions, registrar, personnel, payroll, and athletics. Computerizing the financial aid function can be a difficult task since the information may not be readily available from these departments and must be manually added into the system.

The degree to which other institutional departments are computerized has a major effect upon the financial aid system. If there is little or no computerization in the other departments, the financial aid system may have to be severely limited unless the financial aid office has the resources to input other departments' data into the computer. If other systems are available, programs should be written which transfer their relevant data to the financial aid system (i.e., an interface program). Most ideal, however, is the situation of an institution which has one integrated data base which serves all the departments previously mentioned. If this is available, interfaces are not required.

Financial aid, more than other student services, has a high frequency of updates to a student record. The original offer and the student's response must be put into the computer. Other updates result from need analysis changes. Thus, the student record in the computer is continually being changed. This causes problems in the areas of data entry and turnaround time. The data must be entered into the computer without requiring a tremendous amount of staff time. If extensive coding, batching, and keypunching are required, the amount of staff resources needed will significantly increase. The output (e.g., award letters) must be available as quickly as possible to keep the turnaround time to a minimum. Failure to achieve rapid and efficient data entry can cause a basically well-designed system to be inadequate for the financial aid office.

Many of the data system problems which seem somewhat unique to financial aid can be overcome by using a systematic approach to computer development. A series of developmental steps are needed for the financial aid director to see clearly the scope of the entire project. These steps are as follows:

I. Definition Phase

Metzger in *Managing a Programming Project* (2) indicates that the first phase of a development cycle is the definition phase. Most of the time during this phase should be spent by the financial aid office staff in determining what is needed and should be computerized and in what order. In this phase, operational areas of the office need to be analyzed so that a functional plan can be developed to a high level of specificity. The formulation of this plan should take place over a lengthy period with the constant involvement of a systems analyst and appropriate financial aid staff members. The plan should be put into writing and should be approved by upper management before programming begins.

The plan's chance of success is directly related to the amount of time financial aid personnel are willing to devote to specifying exactly what is needed. This phase should include preparing a written guideline which is followed throughout the project. A well-developed plan gives the aid director the total picture of the project. Mileposts that indicate when something is to be done should be included in the plan by the systems analyst. With time frames detailed, the financial director can be more realistic in terms of what can be expected and when it will occur. The plan should also specify resources needed to accomplish it and possible costs involved. Metzger (2) states, "A list of the problems which most often boil to the surface looks something like this:

- * Poor Planning
- * Ill-defined Specifications
- * Poor Planning
- * Political Pressure
- * Poor Planning

II. Design Phase

Many systems analysts believe design is entirely their function. If the financial aid director has the same opinion, the end result will probably not meet the needs of the financial aid program. Usually there are several solutions to a problem. If the financial aid director is involved in determining the various alternatives, the solution which is most adequate can be better identified. For example, one design might meet the objectives of the plan but might not incorporate the flexibility needed for the ever-changing federal and state regulations.

III. Programming Phase

The system's analyst is responsible for this phase. However, if there are no detailed questions, something is wrong. The programmer is probably making unwarranted assumptions. For example, if the award letter is to be printed only if a freshman is admitted, the system analyst must know what should be done if the student is in an upper class or what should be done if the admit code is blank for a freshman. In particularly crucial areas, the financial aid director should ask the programmer for a detailed description of the logic being incorporated into the program. Only in this way can a financial aid director insure that the program actually does what is intended.

IV. Testing Phase

Involvement by the financial aid director during this phase is helpful to identify any last minute adjustments needed before the program is operational. The importance of this phase must be stressed. A period of parallel operation is highly recommended. After the system is operational it is almost impossible to retract faulty award letters that have been already mailed. The problems caused by inadequate testing cannot be over-emphasized.

V. Operational Phase

Before this final phase is reached, all financial aid staff need to be trained to use the new system. A data processing handbook should be available at workshops for all professional and clerical staff. A batch of student files should be put through the entire process as a training tool. Operational schedules need to be determined and cut-offs established. Adequate back-up procedures are also required.

The managerial effects of having a good system can be varied. Although naive financial aid directors may expect that the computer will decrease the need for staff, in fact computerization usually requires more help for coding, keypunching, and editing. However, the management information obtained is invaluable for completing the Fiscal Operations Report and the Tripartite Application for Student Aid Funds, and for improving knowledge of and control over the financial aid operation.

BIBLIOGRAPHY

- 1 Blumenthal, Sherman C., *Management Information Systems*. Prentice-Hall, Inc., New Jersey, 1969.
- 2 Metzger, Philip, *Managing a Programming Project*. Prentice-Hall, Inc., New Jersey, 1973.
- 3 Optner, Stanford L. *Systems Analysis for Business Management*, Prentice-Hall, Inc., New Jersey, 1968.