The Aspen Institute

MIS for Microenterprise:

A Practical Approach to Managing Information Successfully

by Charles Waterfield

FIELD
Microenterprise Fund for
Innovation, Effectiveness,
Learning and Dissemination

This publication is made possible by a grant from the Ford Foundation.

Published in the United States of America 2002 by the Aspen Institute

Printed in the United States of America
ISBN: 0-89843-357-6
INV#: 02-041

MIS for Microenterprise:

A Practical Approach to Managing Information Successfully

September 2002

Charles Waterfield

FIELD
(Microenterprise Fund for
Innovation, Effectiveness, Learning and Dissemination)
The Aspen Institute
One Dupont Circle, NW
Suite 700
Washington, DC 20036
www.fieldus.org

Acknowledgements

We wish to thank the many people and groups who made this Manual possible. We tremendously appreciate the insights, time and patient review given to us freely by the Management Information System (MIS) Advisory Group who provided guidance on the project that produced this Manual and who contributed substantively to a capacity-building process on MIS for the U.S. microenterprise community. Members included: Marian Doub, Women's Initiative for Self-Employment; Kent Marcoux, Corporation for Enterprise Development; Mary Mathews, Northeast Entrepreneur Fund; Livingston Parsons, ACCION US; Todd Schuldt, Institute for Social and Economic Development; and Tim Shea, University of Massachusetts - Dartmouth. Each of them served as ambassadors for this project to software developers, practitioners and others. They contributed their institutional experiences and their technical expertise, and this document would have been a poorer product without their contributions.

We also express our sincere gratitude to the software vendors who reviewed and commented on the software reviews contained in the Findings Report, attended meetings in California and Washington, D.C., and conducted focus groups in order to better understand the microenterprise market. The vendors all provided timely information on their products and services and showed a great enthusiasm in working with us to improve the supply of products from which practitioners have to choose.

We express our gratitude to The World Bank's Consultative Group to Assist the Poorest (CGAP) for generously allowing us to draw on relevant material from the February 1998 Technical Tool Series No. 1, *Management Information Systems for Microfinance Institutions: A Handbook,* by Charles Waterfield and Nick Ramsing. We also thank Normand Arsenault for useful background information he provided.

Lastly, we appreciate the foresight, support and guidance of Millard ("Mitty") Owens of the Ford Foundation who saw the need for concerted work on MIS for the microenterprise community and supported FIELD's MIS project with the hope of significantly improving practitioners' effectiveness by increasing the products and tools available to them to manage critical information successfully.

Summary Table of Contents

Detailed Table of Contents

Background on the FIELD MIS Project

At a February 2000 meeting hosted by FIELD (Aspen Institute's Microenterprise Fund for Innovation, Effectiveness, Learning and Dissemination) more than 30 microenterprise practitioners participating in MicroTest, FIELD's performance measurement program, came together for their annual meeting to review data on the efficiency, quality and effectiveness of their programs and to consider key issues that influenced the results. During that meeting, FIELD staff, practitioners and Mitty Owens, Program Officer for Economic Development, of the Ford Foundation, reflected on the important role a good management information system plays in supporting a program's ability to collect and analyze the key information needed to better manage its resources and provide services effectively. Participants reviewed the results of a survey of MicroTest members that identified the common difficulties practitioners faced in either finding a suitable software package or developing a high quality Management Information System (MIS) on their own. This conversation led to the creation of a MIS project designed to make a systematic intervention on both the supply and demand sides of the marketplace for MIS products.

This MIS project's goal has been to increase the capacity of microenterprise practitioners to generate key data on program performance and client outcomes to support both better program management and stronger accountability to donors. Understanding that there is not a single solution that will resolve the MIS needs of the great diversity of programs and stakeholders in this field, the project has purposefully focused on market-based approaches to the problem. In this vein, it has worked systematically to increase the acumen of program managers as planners, buyers and users of information systems, and to stimulate better software or database design through clearer communication to interested software developers around the information needs of the microenterprise community. The current Manual, as well as other products discussed below, represents the culmination of learning concerning effective MIS.

Products of the FIELD MIS Project:

MIS for Microenterprise: A Practical Approach to Managing Information Successfully.
This Manual is the second written product to be offered to the microenterprise development community to help practitioners improve their MIS. It builds upon and includes prior work published under the title of Findings Report, which included a detailed review of currently marketed software products for microenterprise organizations.

This Manual pulls together the body of knowledge within individual programs around what information systems need to contain and how they need to operate. It is written for microenterprise program staff and management, and is designed to be an accessible, thorough and practical guide to help users develop an effective MIS for their organizations. It provides step-by-step guidance on understanding:

- the parts of a MIS and their relation to each other,
- what organizational factors influence how a MIS should be designed and managed,
- the process of defining information needs,
- assessing the appropriateness of available software products to meet those needs, and
- how to implement an effective MIS.

2002 FIELD MIS Software Review *(forthcoming October 2002).* This companion document to the MIS Manual is an outgrowth of the Findings Report first published by FIELD in 2001. It contains detailed reviews of major commercial software designed for use in the microenterprise field, applying the evaluative approach described in the MIS Manual. It is our hope that the software review guide will be updated periodically to reflect changes in the software available for the field.

FIELD staff and technical consultant, Charles Waterfield, along with the project's advisors, developed the following additional products as a suite of activities designed to help produce a long-term improvement in MIS for microenterprise programs.

Developer Services. In addition to producing and disseminating this MIS Manual, the FIELD MIS project has endeavored to take a creative and constructive approach to working with the supply side of the MIS market for microenterprise programs, in the hope that clearer communication with software developers concerning the information needs of a diverse community of practitioners will leverage a wider and better product selection to meet those needs. To help bridge some of the information gap between microenterprise program managers and software developers, the project is working with interested vendors of MIS products that are currently being used by micro programs.

Training Workshops. In order to help equip microenterprise practitioners with the skills they need to fully utilize the MIS information in this Manual within their organizations, and recognizing that the microenterprise community has an enduring commitment to learning and peer-to-peer interaction, FIELD offers a one-and-a-half-day, in-depth workshop for practitioners, the first of which was offered in November 2001. Given demand and the availability of continuing donor support for this MIS project, further trainings will be offered.

(For further information about the above products, please contact Jerry Black at the FIELD program of the Aspen Institute: 202-736-2533 or jerry.black@aspeninstitute.org.)

Before You Get Started

Definition of MIS

One of the main messages in this Manual is that when we talk about an effective MIS, we mean more than appropriate software products. Defined in more detail on p. 17, a MIS is:

> *the series of processes and actions involved in capturing raw data, processing the data into usable information and disseminating the information to users in the form needed in order to make appropriate decisions.*

Functions of an Effective MIS

Microenterprise development programs all have unique information needs, as will be discussed below. Every program, however, has to have at least three key kinds of data about its clients.

- Who are our clients and customers? In order to better understand and maintain communication with clients, customers, and other stakeholders, programs often have a demographics and/or a contact management system.
- What services have we provided to them? Microenterprise agencies across the U.S. provide a wide range of services, from different types of business development training programs to loans, from Individual Development Accounts to micro-equity investments. The broader the range of kinds of services a program offers, the more challenging its task in achieving an integrated MIS.
- Why and how are our services important to our clients? Most programs struggle to collect and analyze data that speaks to the effects or outcomes which their services are having on their clients and customers. Monitoring and measuring these outcomes remain challenges in the microenterprise field, and there are fundamental MIS design issues to consider in order to meet these challenges.

The MIS Selection Process

The first step any enterprising microenterprise program manager should take in the journey to obtaining an effective MIS should be clear: Read this Manual.

The next series of steps fall into three main phases: Defining Information Needs, Assessing Alternatives, and finally, System Implementation. These phases, and detailed guidance on how to make progress from one to the next, form the core content of this Manual, beginning on p. 34.

Chapter 1: The Information Dilemma

Information is important, even crucial to the survival of an institution. But, information management is commonly one of the weakest areas of management, and almost certainly one where practice falls far short of theory. This chapter explains what information is and why information sorts are not all alike. It discusses why information management is so problematic and introduces the content of the Manual, suggesting ways in which it can be used to develop your institution's MIS strategy.

1.1. Wasn't the "information age" supposed to solve our problems, not create new ones?

> *"The information you have is not the information you want.*
> *The information you want is not the information you need.*
> *The information you need is not the information you can obtain.*
> *The information you can obtain costs more than you want to pay."*[1]

The above quote summarizes well the ironies of the "information age" and explains why many people consider that this promised golden era has not yet fully arrived. There is plenty of information available, but often it's not usable information. It's either too detailed, too late, inaccessible, or even wrong. Where is this information in the typical microenterprise assistance institution? It is hidden away in file cabinets, scattered about on loan account registers and cashier receipts, buried away in stacks of accounting vouchers, and trapped in the heads of staff, not to mention stored in the vast number of computer data files cluttering up everyone's hard drives — spreadsheets, word processor files and database files.

For most purposes, there is not really a lack of data. What is lacking is that we may not be collecting the *right* data; what is lacking are the systems necessary to *process* the data into usable information; what is lacking is the ability of our staff to correctly *interpret* the information they have and act upon that information. To make information work for us, we need to carefully assess our information needs, determine how we can obtain that information, and invest the resources — human, financial and technical — necessary to have sound management information systems. And, once we have done all that, we need to revisit all of the same issues periodically to see whether our information needs have changed or if there is something new in the ever-improving sea of software products that is, indeed, the better "mousetrap." Information does not come free, but we need to realize that the real cost of information is the cost of *not having* information.

This Manual provides information on how to establish a sound management information system ("MIS") for a microenterprise institution. The issues are many; they are complex; and they are highly interrelated. In many cases, setting up a good information system may necessitate a significant restructuring of the institution, reworking staff responsibilities (sometimes even some staff qualifications), redesigning work processes and information flows, revising and rationalizing financial policies, making significant investment in computer technology — the list goes on. The fact is, information systems can perform only as well as the institutions they model. In reality, information really is at the core of our work; it, therefore, shouldn't be surprising that introducing new information systems can affect the organization to the very core.

[1] Peter Bernstein, *Against the Gods: The Remarkable Story of Risk* (Wiley and Sons, 1996), 202.

In fact, the process and its implications can be so daunting that this may be the principal reason that most institutions have weak systems — they are unable to devote the sustained energy and attention it takes to establishing good systems. However, management in those institutions who have made the investment and now have access to reliable and timely information normally avow that it was one of the most important decisions they ever made.

If you remember nothing else from this Manual, you should remember the following: MIS is not simply a software program. Your institution's information needs are met not only by software, but also by thinking through and rationalizing how your institution collects, processes, transfers and uses information in operational and strategic decisionmaking.

1.2. The importance of MIS to an institution in the 21st Century

Many microenterprise program managers are feeling the need for more detailed, accurate and timely information about all aspects of their work, from training and lending to the effect of their program on the lives of their clients.[2] They know that the better their information is, the more strategically they can manage their scarce resources, and the more they will be able to say about the outcomes of their work. In addition to this internal, managerial desire for high quality information, the microenterprise practitioner community is challenged ever more to report on its performance and to demonstrate impact. Demands come from private and public sector funders, corporate partners and the general public. Who are programs reaching? How well are microenterprise credit, training and technical assistance activities performing? What is the magnitude of investment in programs in relationship to the demand? What are the costs of providing services, and how sustainable are programs becoming? What are the results for program clients and their families?

As managers seek to answer these questions, they find themselves facing several critical constraints. Importantly, the management information systems needed to capture and organize the range of data that programs must process to generate useful information are lacking in significant ways. For example, there are few standard client-tracking systems available "off-the-shelf." The result is that many program practitioners have tried to design their own, with frequent dissatisfaction with the depth or quality of the data that these systems provide, or the difficulty involved in their use. Almost no client-tracking system easily links to loan portfolio management products, and very few adequately capture historical data, limiting their use for understanding client-level outcomes.

Consider the following fictional example of how a program could benefit from an improved MIS:

> *BYOB (Be Your Own Boss) is a young microenterprise development program, housed in a larger non-profit organization that provides a range of services, including refugee resettlement and individual development accounts. The micro program provides a 12-week, 48-hour self-employment training program, one-on-one business counseling and microloans for some of their training program graduates. The program has grown substantially in the past year, due to excellent word-of-mouth referrals and favorable press coverage of their work with very low-income women, including former welfare recipients and refugees. Yet they have to date paid very little attention to their management information system needs.*

[2] This section draws substantially from Charles Waterfield and Nick Ramsing, *Handbook for Management Information Systems for Microfinance Institutions,* Technical Tool Series No. 1 (The Consultative Group to Assist the Poorest, 1998), 1. Future footnotes to this document will refer to the *CGAP MIS Handbook.*

> *Their client-tracking system is an Excel™ spreadsheet that is out of date and becoming unwieldy to use. Staff members have difficulty determining whether a particular client has been through the core training program, or not. As a result, their one loan officer cannot get a clear idea of future loan demand. To complicate matters, the larger non-profit has an individual development account (IDA) program for refugees that is catching on in the community very quickly, and many of these IDA holders are anxiously looking to the microenterprise program for a loan in the near future. Will the current system be able to handle a larger loan program? She is not even sure how to go about evaluating it, but has a suspicion that it cannot.*
>
> *Staff members are starting to worry that the word-of-mouth about their program, which has been so favorable to date, is poised to turn against them. The visionary executive director of the overall organization has been asking for quarterly activity reports from the director of the micro program, who has been unable to provide more than anecdotal evidence and cursory, manually generated information about the number of graduates. Staff members are unable to say with any precision how many clients have gone on to start businesses, the kinds of businesses clients are starting or the number of jobs those businesses create for the community. The program collects a lot of excellent data on its intake form, but is unable to process any of it into usable information about what client characteristics lead to success. Local private foundations have shown some interest in funding their micro work, but are asking questions about their performance and efficiency that are almost impossible to answer.*

An appropriate and responsive information system would transform this program. The staff may be capable and highly motivated, but without good information they will not perform to their fullest potential. A good MIS would enable the program director to be a better manager, providing useful guidance to staff and informative reports to the executive director. Funders would be much more satisfied that the program would manage scarce philanthropic dollars well, and be more inclined to support its growth. Clients would have confidence that they were spending their time with a program that could respond to their needs and help them achieve self-sufficiency.

1.3. Bringing supply and demand closer together

There is clear consensus that good information systems are fundamental to the success of microenterprise assistance institutions. Thus, institutions are providing a "demand" for good software. However, their perception is that the "supply" of software is very limited. Practitioners are generally dissatisfied with the current commercial options, for one reason or another. It is our opinion that there are three main factors on which we must all focus in order to bring software supply and demand closer together:[3]

- *Thorough identification of information needs.* Most managers, field staff, board members — in fact, nearly all information users — have failed to thoroughly document the information needs of their institution. Often, those indicators that they have identified are insufficiently defined to track reliably. The result is that information

[3] The author has developed and published these three points during much of his previous work on this topic, including in the *CGAP MIS Handbook, 6.*

systems are often implemented without a thorough assessment of needs. To address this limitation, this Manual describes a systematic approach for identifying specific information needs and flows.

- *Effective communication between management and systems people.* Program managers and systems developers speak different languages. When compounded by heavy staff workloads and a tendency to compartmentalize operations, the result is often a system that does not meet its users' needs. This Manual is, therefore, aimed at both audiences — information users and system developers. It introduces and explains key concepts and terms from both worlds so that each may better understand the language of the other.
- *Realistic expectations about information technology.* As explained in Section 1.1, information users often wonder why they can't have the information they want, when they want it. Thus, another goal for this Manual is to educate information users about what is possible and how much effort it takes to get good information out of a system.

1.4. Resource development for MIS

If accurate, relevant, complete, reliable, timely, flexible, accessible and secure data is important to your institution, then it is important that management dedicate sufficient resources, on an on-going basis, to developing and maintaining a good MIS. Resources include money to buy appropriate MIS software. (The companion document, ***2002 FIELD MIS Software Review,*** presents information on the price of different commercial systems). But, to restate an important point made above, the price of a particular piece of software, while important to consider, is not the true cost of an effective MIS for your program. Resources include money and people for training, software customization, data collection, data input and implementation. In later sections of the Manual (see Section 4.4.2), we will discuss cost implications of developing an effective MIS. However, based on several years and countless conversations with microenterprise program managers and our own experience, there are still more. An effective MIS begins and ultimately succeeds or fails based on the resource of leadership. Specifically, we recommend that your institution needs to:

- *Make an organizational commitment to MIS.* There is a clear consensus that MIS is not an issue to be wholly delegated to an information technology specialist. An executive director needs to understand her current MIS and what its future capacity needs to be in order to accomplish strategic goals. At a minimum, she needs to spend enough time becoming familiar with her program's MIS challenges, so that she can effectively raise funds to cover the costs of tackling them.
- *Be realistic about what it takes to build an effective MIS.* Managers have to know what they need and why, derive how much it will cost, develop a good pitch for articulating how important it is to their continued organizational and programmatic health, and commit to raising the funds.
- *Build into their annual budget — and fundraising plans — a line item for MIS.* An organization needs good information to survive and grow, just as humans need oxygen to live. Institutions will suffocate from a lack of information, if they are unable to process and use the data (air) they gather (inhale) to make good decisions, to take vigorous steps.

What are some arguments managers can make to raise funds around MIS needs?

- *Recognize that an effective MIS is also in funders' interests.* Funders want to know what happens to their philanthropic dollars. Institutions are all being pushed more and more to demonstrate the effects their interventions have on the lives and communities of people they serve. There are, fundamentally, MIS-related costs to tracking, storing, analyzing and reporting on these effects. But, these costs are not well understood, which can lead to unrealistic demands, over-promising and frustration. The better they understand the MIS-cost implications of donors' demands to demonstrate the impact of microenterprise programs, the better positioned they are to make a compelling argument that those costs should be subsidized with grant dollars.
- *Expect it to take longer and cost more than first estimated.* Nearly every microenterprise practitioner has confirmed that the one lesson they learned from "upgrading" their MIS is that what started out as a small, discrete project to improve some aspect of their MIS turned into something larger, costlier and much more difficult to implement than anyone had imagined. Be assured that building and maintaining an effective MIS is hard work, whether you are a small non-profit, a mid-sized manufacturing company or Wal-Mart. That is why FIELD has undertaken this Manual. If managers spend time with this Manual and absorb the combined wisdom of those who have gone through what their organization is about to undertake, it should help them to save time and money. But, even in the best of circumstances, they need to be prepared for the MIS process to be a costlier, more time-consuming process than initially imagined.

Above all else, managers must remember that *you get out of it what you put in.* FIELD staff heard this comment repeatedly. The same practitioners who warned us about the time and cost of their MIS development also emphasized that in the end their program emerged stronger and better for it. Even if they had known back at the start of the process that it was going to be much more difficult and costly than initially planned, they still would have gone forward. These cautions are not meant to discourage, but rather to encourage managers to go forward, but leading their organizations with their eyes more widely open than they might have been otherwise.

1.5. How to use this Manual

1.5.1. Contents of this Manual

Given the importance of MIS in sound management of an institution, how do we go forth and improve our information management? How do we improve what most managers admit is one of the weaker areas of their institution?

As described in Section 1.3, the keys are thorough identification of information needs, effective communication between management and systems people, and realistic expectations about information technology. The intent of this Manual is to carefully address each of these areas.

Therefore, this Manual hopes to address these issues by increasing the acumen of program managers as planners, buyers and users of information systems, and to stimulate better software or database design through clearer communication to interested software developers around the information needs of the microenterprise community.

The Manual is divided into three conceptual parts:

Part I, **Before You Get Started,** consists of this chapter, "The Information Dilemma," and Chapter 2, "An Introduction to MIS Technology," which describes some important fundamentals about MIS technology directed primarily at non-technical managers, so that they can better understand the issues before them.

Part II, **The MIS Selection Process,** is the bulk of the Manual. It starts with Chapter 3, a brief overview of the process, and then has chapters addressing each of the phases and steps in the process. Chapter 4 describes "Phase I: Defining Needs." Chapter 5 provides an introduction to "Phase II: Assessing the Alternatives." Chapters 6 through 9 address each of the four steps in Phase II. Chapter 10 completes this part of the Manual with a presentation of "Phase III: System Implementation."

Part III, **Putting Your System to Use,** is a single chapter dedicated to "Using your New System" and discusses interpretation of indicators as well as the design and use of reports.

The Manual contains a variety of useful annexes with background information on MIS, supplemental readings, a bibliography and a compilation of the tools presented throughout the Manual.

1.5.2. Using the Manual

This Manual is designed to be used, not just read. It contains worksheets and case studies. It has annexes with resources. And, the companion, ***2002 FIELD MIS Software Review,*** is an effort on our part to do much of the work of Phase II for you, and save you a great deal of effort. (Note that the software review document may be updated regularly in the future to reflect changes in the available selection of commercial software applications. Please check at www.fieldus.org for any updates.)

We recommend you begin by carefully reading Chapters 1 and 2 for a thorough overview of MIS issues. Then proceed with a reading of Chapter 3 to get a sense for the overall process, followed by a brief skim through the remaining chapters. Once you are familiar with the Manual, you can then decide how to go about using the resources to meet your own particular MIS needs. You may be in a position where you need only to make some adjustments in the way you use information in your institution. Or, you may need to enhance your tracking of outcome indicators. Or, you may need to replace the module you use for tracking your training activities. Or, you may be in need of updating your entire MIS, or perhaps even reorganizing the flow of information throughout your entire institution. Whatever your situation, there will be parts of this Manual that will be relevant to your needs.

As mentioned earlier, this Manual has been written both for practitioners and for software developers. Both parties need to better understand the MIS needs specific to microenterprise assistance agencies, and both parties need to strive to improve communication with each other. It is our hope that this Manual helps to contribute toward that process.

Chapter 2: An Introduction to MIS Technology

Section 1.3 identified three keys to successful development of a management information system. This chapter addresses two of them: effective communication between management and systems people, and realistic expectations about information technology.

This chapter is intended to give managers the technical information they need to communicate effectively with systems people and to make appropriate decisions about computerization. It begins by defining a MIS, then describes different technology choices available, and then presents a framework for understanding the "three dimensions of complexity" with which an institution must grapple as it determines a path for its MIS strategy.

2.1. What is a MIS?

So, just what is a Management Information System?[4] For the purposes of this Manual, a management information system is: "the series of processes and actions involved in capturing raw data, processing the data into usable information and disseminating the information to users in the form needed in order to make appropriate decisions."

A MIS is not simply a computer program and it involves more than just calculating numbers. Information management is first and foremost people communicating with one another about events that affect the work of their organizations, so that they may all make appropriate decisions. The chart of accounts, all the forms used by an institution — from receipts to client applications to staff vacation requests — meetings, reports, policies and procedures, the staffing structure, job descriptions, the planning process and, yes, the computer software — all these and more influence the flow of information in an institution and so, together, make up the whole of the management information system.

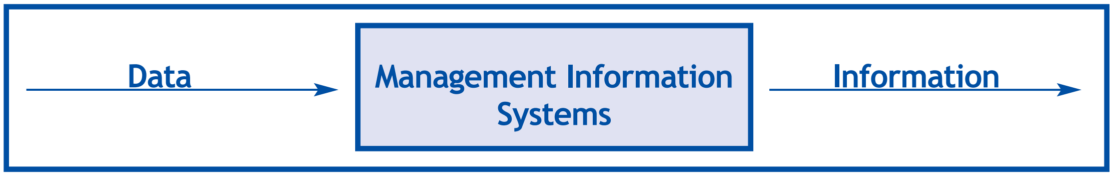
Definition

A management information system is the series of processes and actions involved in capturing raw data, processing it into usable information and disseminating it to users in the form needed in order to make appropriate decisions.

The definition of a MIS makes a significant distinction between data and information.

- "Data" are unprocessed facts that give no insight by themselves. A single loan payment transaction, for example, does not show whether the payment was on time or shed light on the loan's status.
- "Information" is data that is processed or transformed by the MIS "black box" that then helps someone make a decision or gain insight. For example, taking the payment transaction just mentioned and comparing it with prior payments and scheduled installments for this loan reveals the status of the loan and its aging — information that can then be used to make a decision on loan follow-up or loan loss provisioning. The distinction is dramatic. An institution can be swimming in data and yet have little information with which to make decisions.

[4] This section draws substantially upon the description of a MIS presented in the CGAP MIS Handbook, 3.



2.2. Technology choices

Despite the previous section's emphasis on a MIS' *not* being simply a computer program or a piece of technology, this Manual does focus most of its attention on the software aspect of the MIS, as this is one of the weakest areas in the MIS of most institutions. Other aspects of information management — such as organizational structure, staff communications, and annual planning and monitoring — are dealt with extensively in other manuals on business management.

There are three basic technologies that can be considered — spreadsheets, databases, and the Internet or Web. And, within each of the technologies, there are a vast number of options that would require an extremely technical analysis to understand. However, such an analysis is beyond the scope of this Manual because: it would add greatly to its length and complexity; the information would be outdated very quickly because of the rapid changes in technology; and microenterprise managers had best focus their attention on the fundamental issues presented in this Manual rather than attempt to become experts in the nuances of database engines and networking services.

2.2.1. Using spreadsheets

Some microenterprise assistance institutions use spreadsheets (such as Excel™ or Lotus™) to track client information and the delivery of specific services to those clients.[5] These tools are more easily created, altered and maintained than databases, partly because users are more comfortable working with spreadsheets. However, their usefulness is much more limited, especially for organizations handling more than a few hundred active clients. In addition, spreadsheet systems collapse when an institution's structure becomes more complex. For example, it is difficult to consolidate manual or spreadsheet information from multiple branches.

Non-programmers find spreadsheets easy to use because they are essentially computerized ledger books whose formatting and calculations can be easily changed. Spreadsheets were designed to analyze data. But, for storing and retrieving data and reporting on large amounts of data, databases have a clear advantage over spreadsheets because of their data structure (see the following section). Spreadsheets are typically two-dimensional — they have rows and columns — and as a result have difficulty expressing and maintaining complex relationships among data. They maintain these relationships through formulas entered into individual cells rather than key fields that can be rapidly sorted and searched, as in a database. So, spreadsheets are not the optimal tool for recording client transactions and reporting on a portfolio's aging. But, they are extremely useful for analyzing financial and portfolio indicators and for presenting information. In a MIS, spreadsheets can complement the database system managing the bulk of the data.

[5] This section is closely based on the *CGAP MIS Handbook,* 7.

Another useful role for spreadsheets is as an interim tool for conceptualizing or prototyping a database system before it is developed. The individual tables of the database can be built on individual worksheets in the workbook. As mentioned, however, it is quite difficult to create relationships between individual tables when using spreadsheets.

2.2.2. Databases

Databases are the most appropriate computer tool for storing and reporting the financial information that institutions use. They operate well in an institution that depends on a high volume of information and with historical or time-based information. They can generate complex reports from a large data source. Most importantly, databases create functional information systems. They organize information in a system according to its elements (such as scheduled and actual loan payments) and the relationships among those elements. The database structure maintains these relationships through key variables.

As indicated early, managers and practitioners need a basic understanding of database design, so that they can communicate effectively with programmers and system analysts during the design process. They also need to understand the implications and constraints involved in changing the database structure in the future. Fortunately, reaching this level of technical understanding is not nearly as complex as it sounds. Section 2.3 will deal with these issues, providing a basic introduction to database design.

2.2.3. Web-based options

An option with significant future potential is to use a Web-based system hosted by an Application Service Provider (ASP). With such a system, your data resides outside of your institution at the ASP, and you must be connected via the Internet to both input and access your data.

Benefits of a Web-based MIS include the following:

- **Reduced hardware expense.** Most organizations can access a Web-based system without the expense and effort of adding new hardware. No database server is required. Existing PC's or Macintoshes need only be powerful enough to access the Internet.
- **Reduced technical expertise needed on staff.** Networking issues are reduced, especially in multi-location programs. No PC software installation and support are required. System maintenance is significantly reduced. As a result, an organization does not need to devote technical staff to the upkeep of the system.
- **Disaster Recovery Planning is handled for you.** Disaster Recovery Planning anticipates all levels of computer service interruption, from a failed hard drive on the computer to a fire or flood that wipes out a building. The Application Service Provider takes care of Disaster Recovery Plan details. Data are backed up and stored in separate locations in the event of a disaster at the server site. The customer needs not worry about these technical details. Also, if there is a disaster at the customer location (PC disk failure, fire, flood), the organization can be up and running by simply finding another working PC with an Internet connection. (Compare this with the case study found on Page 133.)
- **Immediately benefit from new features and updates.** Enhancements to a PC-based product are only available after the update has been packaged, delivered to the customer

and installed on each PC. When a Web-based product is enhanced, every user benefits immediately without taking any additional steps (such as loading a new program on their PC).

- **Increase program scale quickly and economically.** With a Web-based application, growth planning is greatly simplified. An Internet connection is all that is needed to expand an office or open a new location. A centralized database is accessible by many users from many locations.
- **Benefits can be available to more constituents.** Information can be entered and/or shared with "occasional users." For example, a board member may access reports online, a consultant may enter notes about a consulting session, or a client might self-enroll for a workshop.
- **Aggregate reports are instantly available.** Because data is stored centrally, information for multi-location organizations or for affiliated programs can be rolled together at the touch of a button. For example, a funding agency can instantly see a report of outcome information for all programs it supports.
- **Marketing tool.** A Web-based database can become an integral part of an organization's Web presence. Increased ability to serve constituents provides a great marketing tool to the organization. In addition, ASP's often include a Web site for your institution with its own unique Internet name (URL). This could serve as your institutional Web site. Fairly easy-to-use Web-page — development tools like Microsoft Front Page™ are often part of the services the ASP provides.

There are, however, some potential drawbacks of relying on a Web-based system. These drawbacks need to be carefully considered when making a decision to use an ASP.

- **Confidentiality issues.** Client information is not stored on-site. This may be an issue for some institutions' policies or the policies of their funders.
- **Dependence on the ASP.** If you purchase software and the company goes out of business, you can continue to use the software (unless there are access codes that need to be renewed). If you use Web-based software and the ASP goes out of business, you can no longer access the software or your data. This is obviously a critical issue and you need to carefully analyze contingency issues with your ASP. Most vendors do have a contingency plan in place to deal with this very real possibility. Also, be sure to carefully read the termination language in any contract. Be sure that your data cannot be held hostage if you decide to end your relationship with the ASP.
- **Web-based response times.** Information must be processed by the server and transmitted over an Internet connection. This can potentially result in sluggish performance.

Currently, the marketplace for ASP is large and growing fast. And, costs for a useful service are quite affordable — a service with a reasonable set of services might range from a few hundred dollars a year to a few hundred dollars a month. However, this marketplace is new and dynamic and should be approached with a "buyer-beware" attitude.

2.3. The three dimensions of complexity

Certainly information management is a critical need for an institution. Yet, deciding how to successfully satisfy that need is far from easy. Even in the apparently straightforward process of selecting “off-the-shelf” commercial software, an organization should first engage in a period of self-analysis. Two questions should be answered in detail. In systems analysis parlance, the two questions are “what is?” and “what should be?”

- **What is?** How do we do our work now? What steps do we take? These questions can be documented using tools such as process diagrams and data flow diagrams.
- **What should be?** With a full understanding of how an organization currently does its work, improvements can be built in by reworking existing processes and building in new processes.

After these sometimes time-consuming steps are completed and an organization is very clear what their new software system needs to be able to do, then the time is right to investigate what “off-the-shelf” software is available. Fortunately, we live in a time when the number and quality of “off-the-shelf” software packages are growing. Unfortunately, as you will see below, that process of matching the right software to the right situation is not that simple.

The issue is difficult because MIS software is designed to mirror the operational procedures and informational needs of the institution, and there are no two institutions with exactly the same operational procedures and information needs. Every institution is different, in most cases *very* different. So the “fit” between the software and the institution is imperfect at best. A general litmus test is the 80/20 test — if the software package can do 80 percent of what you need, then buying and customizing that package is usually preferable to developing a package from scratch. The software selection process is essentially a search for the most reasonable fit, a sort of decision to marry a “compatible” partner as you abandon your search for your “soul mate.”

If we understand why this imperfect fit exists, we can use that awareness to improve our decisionmaking. The fundamental issues affecting software compatibility for microenterprise agencies can be grouped in three distinct areas:

1. The **breadth,** or diversity, of services provided. In microenterprise assistance this can include credit, IDAs, training, technical assistance and other services.

2. The **depth,** or complexity, of information needs. Different institutions have different needs and preferences for the amount and complexity of the information they track, particularly in the areas of demographic data and in measuring outcomes, i.e., the impact their services are having on clients.

3. The **scale** of the institution. As an institution grows in staff and clients, its information needs will change in substantial ways, primarily with regard to capacity and security issues.

These three areas can be presented as three layers to portray how complex the MIS needs to be. A straightforward but useful means of using this system is to evaluate your institution’s needs in each of the three areas on a simple scale of basic (one point), intermediate (two points) or advanced (three points). Thus, possible scores range from a minimum of three (scoring one point in each area) to a maximum of nine (scoring three points in each area). It must be emphasized that there is a significant difference in complexity in even a one-point difference. It may only be a slight exaggeration to compare this scoring system to the Richter Scale used to measure the magnitude of earthquakes, where an increase of one point indicates a ten-fold increase in the power of the earthquake.

Institution "A"				
Category	Basic	Intermediate	Advanced	Score
Breadth	X			1
Depth	X			1
Scale	X			1
Total Score				3

Take the example of Institution A, which has only one service to track (training), has minimal information needs (it primarily tracks course attendance information, needs very little demographic data and does not monitor outcomes), and operates at relatively small scale (about 100 clients served by three staff). It scores a total of three points, the minimum score, indicating that it should be able to find a reasonable solution to its information needs. It should be able to either use a basic commercial package or even be able to develop its own homemade solution by using a product such as Microsoft Excel™.

Institution "B"				
Category	Basic	Intermediate	Advanced	Score
Breadth	X			1
Depth		X		2
Scale		X		2
Total Score				5

Institution B, a credit union, provides only loans (scoring one point for "breadth"). The institution must track information for businesses with multiple owners as well as owners with multiple businesses, and they are required to track detailed household income levels to satisfy donor requirements, thus scoring two points for "depth" of information. Finally, they are of medium scale and need to have a networked system with reasonably good security measures built into the software, such as user logons and protected data files, thus scoring two points for scale/sophistication. Their total score is thus five, indicating that their task is considerably harder than that for Institution A.

Institution "C"				
Category	Basic	Intermediate	Advanced	Score
Breadth			X	3
Depth			X	3
Scale			X	3
Total Score				9

Finally, Institution C offers multiple services (credit, IDAs, training courses, individual technical assistance and referrals to other social agencies), its information needs are complex (it tracks a large number of demographic indicators for its multiple donors and is committed to monitoring outcomes), and is of significant scale (e.g., 1,000 clients served by 20 staff) and is concerned about issues of client confidentiality. The resulting score is the maximum possible of nine, indicating that this institution has a very significant task and will need a much more advanced approach to information management. It will likely need to expend a great amount of time, effort and money in finding and implementing a solution to its information needs. However, the benefits from undertaking such a MIS development effort are often greater, as well. A manual or ad-hoc computer solution in an environment of complex information needs is typically even more cumbersome and unreliable than in less complex information environments.

The following sections provide more detail to these three dimensions and the resulting implications for designing a MIS that meets the institution's needs.

2.3.1. Dimension 1: Breadth, or diversity, of services offered

Institutions involved in microenterprise assistance often are involved in a variety of services, not just lending. While organizations may be training- or credit-led, they often offer credit, training and technical-assistance services. They may offer access-to-markets services, such as promotion or brokering. They may offer additional financial services, such as Individual Development Accounts. This creates profound challenges for seeking MIS solutions, as the institution faces the issue of assembling and linking various modules.

MIS modules

MIS software is generally comprised of modules that are linked behind a common user interface. The modules share certain tables of common information, but the information specific to each module is stored independently, with the common interface managing the linkages among the modules. Take the example of accounting and loan-portfolio management. Most of the portfolio module is actually self-contained and distinct from the accounting module. Generally, portfolio information is summarized and input into the accounting system automatically at discrete moments in time, e.g., when "batches" of information are "posted," or when close-of-day routines are run.

Linking modules

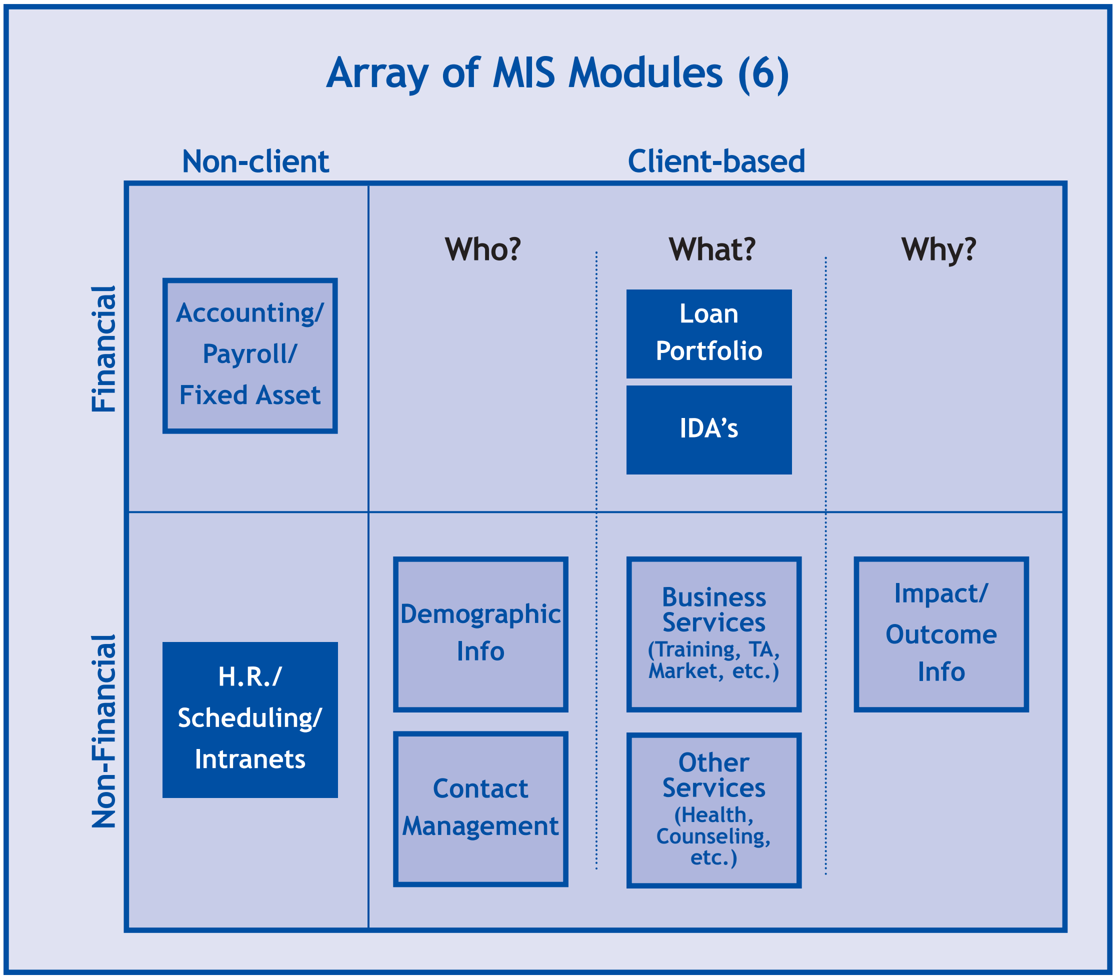
If an institution chooses an accounting module from one vendor and a loan portfolio module from a different vendor, it is highly unlikely that the two modules will transfer information, i.e., interface, automatically. This creates extra work for program staff, which must manually tabulate information from the loan module and enter that information in the accounting system, generally as part of a close-of-day process. In addition to the additional work, there is the risk that errors will be introduced and the two systems will get "out-of-sync." The good news is that most commercial software packages today offer various means of sharing, or importing/exporting data. The bad news is that the process is often not as simple as the manual that comes with the software claims, and will typically require the careful work of knowledgeable staff. In fact, this may be a prudent time to hire a consultant to initially set up the transfer process.

Also, management reports may require manual preparation, extracting certain information from each module and presenting it on an overall report. Still, such a manual approach to linking information and generating reports may be a reasonable cost; the most appropriate software for your needs is not necessarily the software that is fully integrated.

Additional guidance on linking modules can be found in Section 9.2.2.

Array of potential modules

The more diverse the array of services and activities performed by an institution, the greater the need for additional modules to monitor and manage the data generated by those activities. In other words, a multi-service institution will need to employ more modules in its MIS.



Array of MIS Modules (6)

Non-client
Client-based
Financial
Non-Financial
Who?
What?
Why?
Accounting/ Payroll/ Fixed Asset
Loan Portfolio
IDA's
H.R./ Scheduling/ Intranets
Demographic Info
Contact Management
Business Services (Training, TA, Market, etc.)
Other Services (Health, Counseling, etc.)
Impact/ Outcome Info

The above graphic presents a conceptual grouping of the potential modules required by microenterprise agencies. The framework is divided into four major sections. The horizontal line differentiates between financial information and non-financial information. The primary vertical line divides between modules that are client-based and those that are not client-specific.

All institutions need accounting software, which may include management of payroll, fixed assets, accounts receivable/payable and other financial areas. There is also a need, not necessarily software-based, for tracking personnel, managing schedules and sharing information internally. Regardless of the system used to manage this information, it can still be considered a part of the overall Management Information System.

For institutions working directly with clients, there is a need to track information at the client-level. The needs in this area can be differentiated into three major components: the **who, what** and **why** of service delivery.

- **Managing the "who"**

The purpose of the "who" portion is to better understand and serve the clients. This is accomplished by collecting **demographic information** on the client. Depending on the interest of the institution and its stakeholders, this area can be minimal or quite extensive. For social service institutions, the type of demographic information collected will depend on the specifics of the target group, the requirements of funding agencies and the interest in correlating socioeconomic data with outcomes tracked in the impact/outcome module (distinct from, but related to the demographic information module). Note that in many instances, the institution will find it valuable to track outside contacts as well as clients — community-based organizations, media, donors, friends, volunteers, etc. This will require either a separate module or a means to distinguish between clients and non-clients.

Some institutions are interested in **contact management** software as a means of improving the quality of its services and the productivity of its staff. Such software is client-centered and provides a quick and efficient means of logging all contacts that take place with the client — phone calls, office visits, field visits, class attendance, etc. Information generally logs the date and time of the contact, the duration, the staff person making contact, the type of contact and a memo field for recording notes and observations. Contact management software can also be used to schedule future planned appointments.

- **Managing the "what"**

Critical modules for an institution are those that track the "what", i.e., the services provided by the institution. There is a wide variety of services that can be offered by institutions involved in microenterprise assistance. Financial services modules consist of **loan portfolio** modules and possibly Individual Development Account, or **IDA,** modules. Although both modules monitor financial activity, it is not necessary for them to be fully integrated with the accounting system; as explained earlier, this integration can be managed by manual processes.

Non-financial services can include **Business Development Services,** such as training, technical assistance and marketing. Although there is not the same strict need to be financially accountable as with a loan portfolio, an institution will consider it a matter of institutional accountability to monitor the volume of services it has delivered to its clients.[6] If the volume of services and the number of clients are small, information may be managed through simple, paper-based systems or a basic spreadsheet. Larger institutions will seek a more sophisticated database solution.

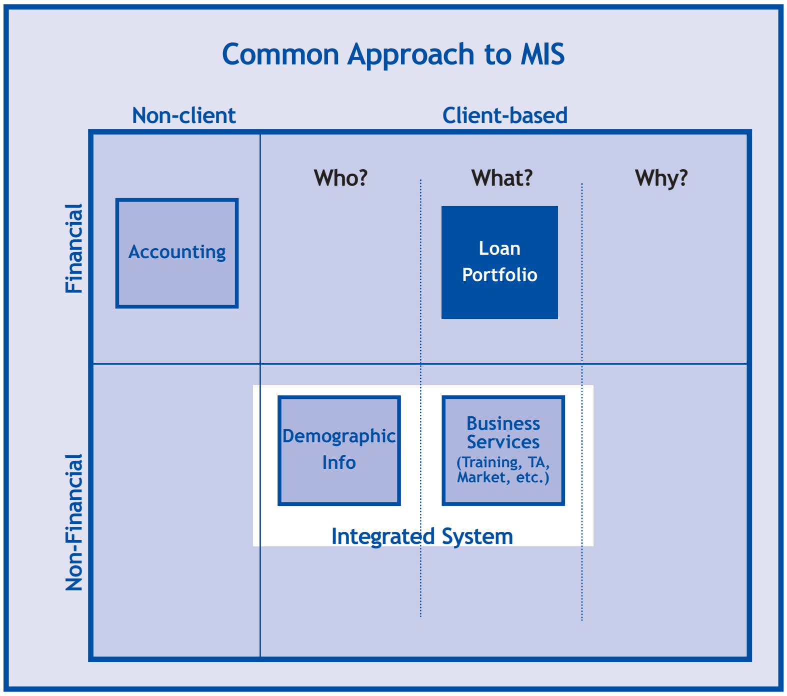
Some integrated, multi-service agencies offer **Other Services** not related to the client's business, such as health services and counseling. The more diverse the institution's range of services, the greater the challenge to find integrated modules in a commercial system. Finding systems that combine microenterprise and non-business services is quite uncommon.

- **Managing the "why"**

Microenterprise assistance agencies have a strong desire, even a need, to monitor and measure outcomes resulting from the services they provide to clients. This is an area that is still undergoing study. There is not yet consensus on the extent to which outcomes measurement should be standardized.[7] In the absence of such agreement, the best solution that software companies can provide is an open-ended system that lets the user identify the data points to be monitored. This is actually not that difficult to implement in a modest fashion, and hopefully commercial products will emerge to satisfy this market demand.

[6] Note that costs and cost-recovery of business development services will be tracked through the accounting module. The primary concern here is with tracking the amount and type of services received by clients, e.g., person hours of technical assistance or classroom hours of training, by topic, as well as issues such as completion rates.

[7] It should be noted that there is a continuum of outcomes measurement, from follow-up monitoring of basic outcomes that happens at regular intervals, to more in-depth outcomes evaluation that happens with a sample of clients and is usually a special effort receiving special funding. The latter requires an investment in evaluation design and sampling expertise.



Assembling the modules into an information system

An institution needs to select an array of modules to tie together into a coherent system that meets its overall information needs. As should be evident by this point, this is not an easy task, particularly for a multi-service institution.

Except in rare cases, the modules are not linked, but must be separately maintained by the agency's staff, and management reports must be manually assembled by extracting information from the various systems. Often the resulting MIS is a mix of commercial software supplemented by custom software developed in-house, supported by a strong foundation of reliable hardware.

Other complications of maintaining non-integrated systems include the potential of different client identification numbers, the inability to source all data on a specific client quickly, and the difficulty of duplicated information. For example, basic demographic data will be requested by each module, particularly name and contact information. The institution needs to decide on which system they will maintain up-to-date information, i.e., when a client moves, which system will hold the current information.

Workable solutions to MIS will vary from institution to institution based on an analysis of its services, its information needs and its scale, which is then compared with the range of commercial offerings and an analysis of the institution's internal ability to develop and maintain a system. The following box provides examples from three institutions about how they have approached the modules issue of MIS.

Examples of MIS Module Integration

ACCION[8]	
Accounting	GMS™
Loan Portfolio	GMS™ with custom modifications
Contact Management	Custom Microtest Access™ database
North East Entrepreneur Fund	
Accounting	GMS™
Loan Portfolio	GMS™
Training and TA	Custom Microsoft Access™ database
West Company	
Accounting	QuickBooks Pro™
Loan Portfolio	Down Home Loan Manager™
IDA	MIS IDA
Client Management and Training	Custom Microsoft Access™ database

2.3.2. Dimension 2: Depth, or complexity, of information needs

Depth of information refers to the amount of detail to be collected and stored by the system. Depth of information can be portrayed by contrasting two institutions and their needs

[8] ACCION is currently in the process of replacing this MIS configuration with new, integrated software.

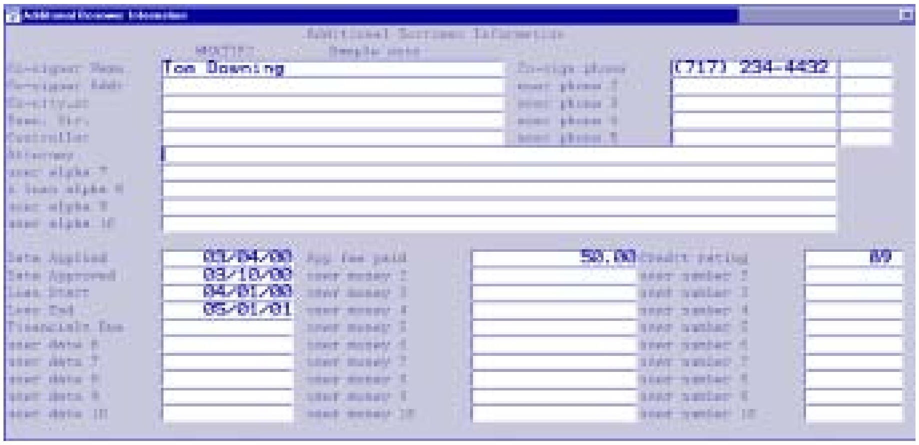
for socioeconomic information. Institution A collects a basic set of information: gender, race, age, marital status, household income, citizenship and whether the client has received welfare assistance in the past six months prior to entering the program. All of this information is stored in static form, i.e., information is input at the time the client registers, and if any data changes, such as marital status or income level, the software replaces the information stored in the system with the new information. That is, any historical data is not stored, but is lost forever, unless stored in some paper-based format. Most software employed by microenterprise agencies stores information in this static manner, oftentimes with information users unaware of the fact.

Institution B has much greater socioeconomic data requirements. In addition to the basic data points tracked by Institution A, it needs to track the relationships of household members, household income needs to be disaggregated by household member and source, and changes in income for each member need to be tracked over time. The institution also needs to track quarterly balance-sheet and income-statement data for the clients' businesses. This level of depth applies to a number of other socioeconomic data points as well.

Clearly, Institution B has significantly greater information needs than Institution A. Whereas Institution A is likely to find the basic information collected by any of the commercially available software to be adequate for its needs, Institution B will find most, if not all, available software to fall far short of its expectations and will need to develop customized software based on its own detailed specifications.

Three common means of addressing depth of information needs in software development are the incorporation of user-defined fields, configurability options and the distribution of information among related tables (database architecture). These are discussed, in turn, in the following paragraphs.

User-Defined Fields in LoanPro™



Tom Downing

(717) 234-4432

03/04/00 | 50.00 | 89

03/10/00

04/01/00

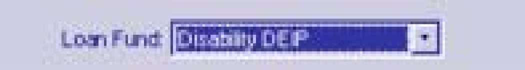
05/01/01

User-defined fields

Software companies cannot anticipate 100 percent of the data points an institution will want to track. One solution to this is to incorporate user-defined fields (UDFs), i.e., fields that the user can name. The software will provide a customization screen that allows the user to name the UDF. This name will then appear on data entry screens and printed reports.

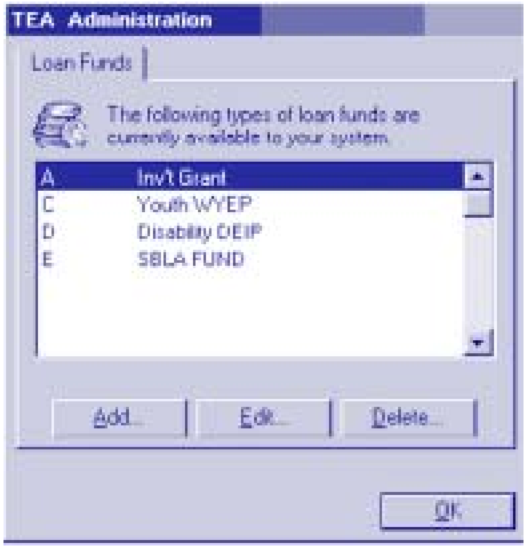
Fields in a database are designed to hold a specific type of information, e.g., text strings, numerical values or dates. Therefore, for increased flexibility, the system should incorporate UDFs of each type. However, UDFs do not always provide the means to use enhanced features such as radio buttons, checkboxes and dropdown lists to select field values. Thus, there is a greater risk of data entry error. For example, a text-based UDF requires the user to type in a response. If the field is used to identify the staff person assigned to this client, the name might be entered or spelled differently in each instance. This restricts the ability to use this field to either sort or filter the information.

TEA Dropdown List



Loan Fund: Disability DEIP

TEA Dropdown Configuration Screen



The following types of loan funds are currently available to your system.

A	Inv't Grant
C	Youth WYEP
D	Disability DEIP
E	SBLA FUND

Configurability

To enhance database integrity, i.e., insure that data is entered properly, user choices are often restricted by means of selecting one or more pre-defined choices on the screen, displayed either as radio buttons, checkboxes or dropdown lists. The available choices must either be anticipated by the software developer or the software must provide a means for the user to customize these lists.

Flexible programs provide a configuration menu that allows the system administrator access to the tables used to populate dropdown lists, where records can be added or deleted and descriptions of each record may be modified. An example might be the staff person assigned to the client. If the staff person is identified in a UDF, as indicated in the previous section, there is risk that the name will be input inconsistently. However, if the staff person data field is linked to a customizable dropdown list, then data will be quickly and consistently entered. The software can then provide an option to use this field as a means to sort reports by staff person or to filter and print out a report for that specific staff person.

Database structure

A database has five basic parts:

- **Tables** are used to store data;
- **Forms** allow users to enter, edit and view data;
- **Queries search** and report data;
- **Reports describe** the format for paper or screen output; and
- **Programming Commands** make it possible to customize a database to fit a particular system.

Arguably the most important decision in database development is the layout of the database tables, or database structure. The database structure defines what information can be stored in tables, as well as the relationships among tables (a crucially important concept, as will be shown). Thus, the database structure is like the foundation of a house because all the forms, queries and reports are built upon it. If the database structure is poorly designed, users will have trouble storing or retrieving information.

Thus, database structure entails deciding the number of tables used in the system, what information is to be stored in each table and how those tables are to be related to each other. These decisions are made very early in the software design process, before entry screens and output reports are designed, and before the source code is written. These decisions must be based on a very thorough understanding of the informational needs of the user. The choices made at this early stage determine to a great extent what the final software will or will not be able to do, and may even limit the ability to properly modify the software at a later stage to accommodate changes in information needs.

Managers need to understand this fundamental issue as they decide on the software they will use to manage their information. This doesn't mean they need to become database design experts, but that they do need to understand the implications of different designs and to know the right questions to ask when they evaluate a software package. Managers also need to be aware that prudence may dictate hiring a database expert for a short period in order to seriously evaluate the database architecture underlying a software package — both in terms of its current design and its flexibility in regards to modifications. This software is likely to become a core component in the daily operations of an agency; making sure the database architecture is robust, is akin to being sure the foundation of a house is done correctly before building the rest of the house.

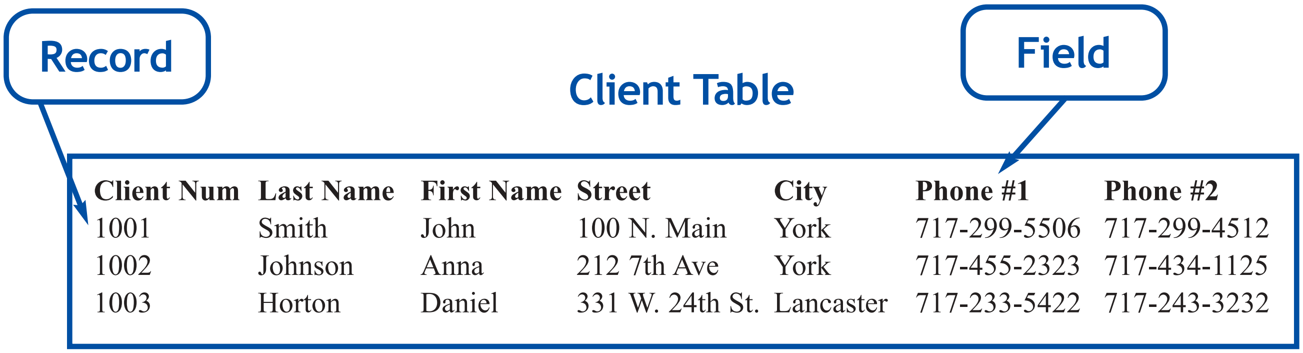
The following sections describe the fundamentals of how database systems store information.

Database design fundamentals

Databases store information in **tables,** which are comprised of **fields** and **records.** Tables, fields and records are analogous to a simple spreadsheet, in which the data points the user wishes to store are defined in columns (i.e., fields) and new data is added by the inclusion of new rows (i.e., records).

Early in the design process, the systems engineer needs to identify what information the user needs and then to design the tables where that information will be stored. This is where the critical decisions are to be made, resulting in systems that are typically either limited in scope but simple to develop, or highly detailed in content but quite complex to develop and maintain.

We can better understand this trade-off by reviewing two approaches to the same situation. The first approach is a basic solution to the problem of storing contact information for a client. The engineer decides to store one address and two phone numbers for each client, and designs the following table:



Client Num	Last Name	First Name	Street	City	Phone #1	Phone #2
1001	Smith	John	100 N. Main	York	717-299-5506	717-299-4512
1002	Johnson	Anna	212 7th Ave	York	717-455-2323	717-434-1125
1003	Horton	Daniel	331 W. 24th St.	Lancaster	717-233-5422	717-243-3232

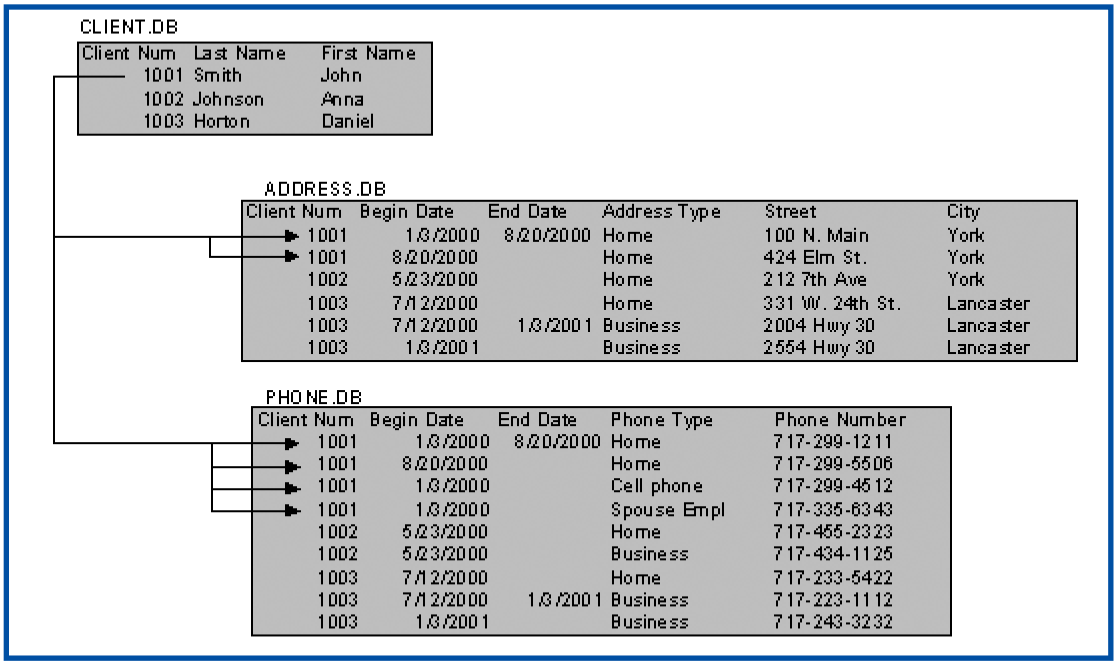
This straightforward approach resembles a spreadsheet and could, in fact, easily be created and maintained on a spreadsheet. Contact information resides in a single table, where it can easily be viewed and printed. Design of the database and the entry screens can be done in minutes. However, the design has significant limitations that may limit its usefulness to some users.

First of all, the structure limits the number of addresses and phone numbers that can be stored. This limitation could be addressed, within limits, by adding more fields to store more options, e.g., adding a "Phone #3" field. With commercial software, however, this approach is limited in that the designer does not know in advance the number of fields the user will desire. To build in a large number of additional fields for every instance of data (phone number is but one simple example) is not feasible. Thus the user needs to live with the decisions made by the designer.

There is a second limitation to the basic approach as well. When a client's contact information changes, it needs to be changed in the database. With this simple table structure, the data currently in the table is *replaced* with the new information. All historical data is lost.

This may not be considered crucial for contact information, but there are many other data points where changes in status or value are important for informational purposes, such as measuring outcomes or generating program statistics. In addition, storing changes in data points is helpful for resolving data input errors (e.g., the data entry person changes the phone number for the wrong client, losing the client's correct phone number) and even for malicious behavior (e.g., an irate employee deletes all client contact information on his last day of work.)

The system engineer's solution to this dilemma is to store the information in multiple tables. Compare the alternative structure in the following figure:



CLIENT.DB

Client Num	Last Name	First Name
1001	Smith	John
1002	Johnson	Anna
1003	Horton	Daniel

ADDRESS.DB

Client Num	Begin Date	End Date	Address Type	Street	City
1001	1/3/2000	8/20/2000	Home	100 N. Main	York
1001	8/20/2000		Home	424 Elm St.	York
1002	5/23/2000		Home	212 7th Ave	York
1003	7/12/2000		Home	331 W. 24th St.	Lancaster
1003	7/12/2000	1/3/2001	Business	2004 Hwy 30	Lancaster
1003	1/3/2001		Business	2554 Hwy 30	Lancaster

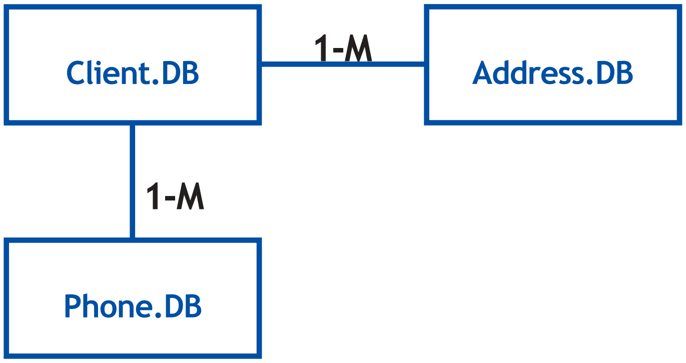
PHONE.DB

Client Num	Begin Date	End Date	Phone Type	Phone Number
1001	1/3/2000	8/20/2000	Home	717-299-1211
1001	8/20/2000		Home	717-299-5506
1001	1/3/2000		Cell phone	717-299-4512
1001	1/3/2000		Spouse Empl	717-335-6343
1002	5/23/2000		Home	717-455-2323
1002	5/23/2000		Business	717-434-1125
1003	7/12/2000		Home	717-233-5422
1003	7/12/2000	1/3/2001	Business	717-223-1112
1003	1/3/2001		Business	717-243-3232

Contact data fields have been removed from the Client.DB table and stored in two separate tables, one for addresses and another for phone numbers. The client table contains the name of the client and a unique client number field, called a **key.** This key is used to "relate" records in one table with records in another table belonging to the same client.

The information system is now considered a **relational database.** Different relationships can be established among tables, depending upon the use of keys. In the above example, the client table is related to both the address and phone tables in a **one-to-many relationship,** i.e., any one record in Client.DB may have an unlimited number of records in either the Address.DB or Phone.DB tables. This is illustrated graphically in the simplified representation on the next page by the markings "**1-M**". These records can be identified via the "Client Num" key. It is unnecessary, and actually inadvisable, to repeat the client's name in either of the other tables. The user's search starts in the Client.DB table. Once the client's name is located, her client number is used to locate her current contact information, which can then be displayed for the user.

Note that this new structure solves both drawbacks of the single-table approach. Firstly, an unlimited number of addresses and phone numbers may be stored. A new field in each table, "Address Type" and "Phone Type" is used to identify whether the information belongs to home, business, other family member, etc. Secondly, obsolete contact information need not be deleted or overwritten. By the addition of two fields, "Begin Date" and "End Date," each record can be marked as to its active status at any given date in time.



The drawback to this solution is an obviously greater degree of complexity in the programming necessary for data input screens and report generation. In some cases the increase in complexity is justifiable; in others, the limitations of the basic approach can be tolerated. The important issue is that managers understand the structure of the system under consideration, identify its limitations and decide if the system is suitable for the institution's needs.

The area in which many commercial systems are weak is the area of outcome monitoring. Measuring outcomes generally requires the storage of periodic data, such as household income before receiving assistance and at periodic intervals thereafter. The data is best stored in a table that allows a record to be added each time income is surveyed.

In summary, the depth of information the institution chooses to monitor has significant implications for the complexity and sophistication of the software application. Basic needs can be satisfied with spreadsheets, or even paper-based systems. Intermediate needs require relatively "flat" database systems with a limited amount of customization. Advanced needs require a large number of tables, connected through complex relationships of keys, paired with a high degree of customization via user-defined fields and customizable lists.

NOTE: Annex 3 contains a helpful exercise for applying this information. It presents the table structures for a simple system and allows you the opportunity to determine the strengths and limitations of the system.

2.3.3. Dimension 3: Scale issues

The third dimension of complexity to consider is the issue of institutional scale, in particular, how scale impacts the technical sophistication of the implementation and ongoing maintenance of the system. Scale affects a number of issues with respect to both software and hardware, including security considerations, network issues, and the need for in-house MIS support staff and potential hardware upgrades.

Security issues

In a computerized information system, security means protecting data, restricting user access and limiting fraud through control routines. Security does not guarantee that fraud or data loss will not occur; it only seeks to minimize it.

System security issues include password entry to user menus, maintenance of user logs, multiple user access levels, client confidentiality issues, protected database formats, locking of data posted and audit trails. Small institutions with only a few staff accessing the software can tolerate a lack of system security, but larger institutions require more rigorous systems.

More secure software maintains a database of authorized users. Each user is assigned a password and an access level. Low-level access is typically for reference purposes only. At increasingly higher levels, users are allowed to enter new data, modify existing data and change system settings.

Advanced software maintains a user log of who was using the system and during what periods of time. Advanced software also may record the user's identification and the date and time of any data entries or modifications. The software may also maintain a copy of any alterations to data. Together these two features provide an "audit trail" to trace back, and detect and correct any errors introduced into the data.

Particularly with financial data, more secure systems will permanently lock data that is already posted. If an error is later detected, it must be corrected through an adjusting entry that explicitly reverses the error. This again serves an audit trail function and also avoids introducing changes that throw off reports generated for previous periods of operation. As an example, QuickBooks™ is *not* a secure system, as users can go to any period of time and introduce new data or modify or delete existing data. Peachtree™ accounting *is* a secure system, in that any entry posted to the books is permanent, and errors must be corrected through introduction of new postings.

A MIS can have these security features incorporated into the program and still not be entirely secure. Logons and passwords only affect access through the system's menu structure. In many, primarily older systems, it is a straightforward process to simply open up the data file and alter the contents, e.g., Microsoft Excel™ can be used to open up older Dbase-compatible files (DBF files) without logging on and using the MIS software. Newer systems have means to password protect the data file.

A final area of security concern is client confidentiality. Social service agencies may need to restrict access to certain sensitive client data, e.g., medical history, to specific staff. This level of protection goes beyond generalized user access levels and actually needs to block information from viewing except by a specific, authorized person.

Networking

Networking issues include allowing multiple concurrent users, as well as connecting and sharing information with different offices.

Except in the smallest institutions, it will be advantageous to employ a multi-user software application that allows more than one staff person to consult and/or enter data at any given moment. Note that you can have a MIS installed on your network and still not be able to have multiple users access data at any one time. A useful analogy is to compare a single-user MIS residing on a network to a word processing document residing on a network; all network users have access to the document, but if any one user has opened the document for editing, all other users are blocked from editing at that time.

Multi-user databases used to be fairly complex programmatically, as the application needed to insure that if two users were logged on to the system, they would not be modifying any one specific record at the same moment in time. In current database development languages, "record locking," or giving permission to modify a record to only one user at any specific moment, is a straightforward process.[9]

[9] In reviewing commercial software, note that a number of the older applications still being sold are *not* designed as multi-user applications, even though you may be able to install the application on a network.

Multi-user databases can be installed on Local-Area Networks (LANs), meaning all users are on computers typically connected within the same physical office, or on Wide-Area Networks (WANs), meaning users are connected to the actual database files over a dial-up network, such as a phone line. If an institution has decentralized operations, it will need to either apply a WAN or it will need to develop a system of independent databases, maintained at each office, that are at some point aggregated to provide information for the institution as a whole. Either alternative requires an extra layer of complexity in both installation and maintenance of the MIS. Also see Section 2.2.3 for a relatively new and cost-effective option, using a Web-based system hosted by an Application Service Provider (ASP).

MIS technical support

Technical demands of the software determine if the institution needs ongoing computer technicians to keep things running smoothly. The more complex the system, the more need there will be for in-house — or contracted, out-sourced — technical staff to maintain the data files, to adapt the system to evolutions in the institution's information needs, to design new report formats and to extract data for special evaluations. In addition, complex software will mean that the technical personnel will need to be involved in staff training and in maintaining computer hardware.

This is a critical area to consider when selecting software. Refer to Section 7.4 for further information.

Hardware upgrades

In many cases, installation of advanced MIS software will require substantial investment in new hardware, operating systems or other software. Hardware costs can easily match or exceed software costs and should not be overlooked when preparing the MIS budget.

See Section 10.2 and Annex 5 for additional information on hardware decisions.

2.4. Conclusion: Dealing with complexity

As should be apparent at this point, deciding how to satisfy an institution's information needs is far from easy. The temptation often exists to "rush out, buy some software and get back to real work." This approach always fails, leaving only frustration in its wake. The proper approach always starts with a needs assessment. Study the institution from the perspective of the three dimensions of complexity outlined here. Get a sense for the size of the challenge ahead. No institution faces an easy task, and the higher the score, the bigger the challenge.

There is no one solution that will work for every institution. There is no ideal solution that will work for any one institution. This failure to achieve perfection is not due to poor management decisions or to unprofessional software developers. This failure is due to the nature of the problem. Information management is inherently difficult. The largest businesses, with vast financial and human resources, struggle with their information systems. How can we expect small non-profits spending only a few thousand dollars to fare any better?

Fortunately, the situation is improving with time and experience. The gap is gradually closing between reality and the ideals to which we aspire. Hopefully, the products that comprise the FIELD MIS project — educating practitioners on the issues, analyzing and documenting existing software, and working with practitioners and vendors to expand communication and understanding of each other's needs — will facilitate closing some of that gap.

Chapter 3: The MIS Selection Process

The core content of this Manual focuses on the process of selecting a MIS, and this chapter serves to introduce that process. Subsequent chapters serve to elaborate the individual steps in the process.[10]

3.1. Overview of the selection process

The MIS selection process is portrayed in the figure on the next page and is divided into three key phases: Defining Needs, Assessing the Alternatives and System Implementation. Each phase is critical to a successful outcome. The process may look tedious and unnecessarily complex, but careful adherence to the principles highlighted here has been proven to work, whereas institutions looking for shortcuts have very often met with failure.

All experts agree that the only place to begin is with an accurate and thorough analysis of the information needs for the institution. A systematic approach for doing so is described in the following chapter, explaining: who should be involved in such a process; what type of information needs documentation; what sources to use; how to anticipate future needs; and how to temper the desire for information with the realities imposed by limitations in staff capabilities and financial resources.

This clear understanding of the institution's information needs is essential in order to begin Phase II: Assessing the Alternatives. Some software is just plain bad. But, most software applications are well-done, professional work. The fact that an application is well done, however, does not mean that it is suitable for your institution's needs. Therefore, the principle goal of this phase is to assess the "fit" between your institution's needs and the various software alternatives under consideration. This process can be challenging because making a sound decision depends not only upon having a clear understanding of your needs (as you will have developed during Phase I), but also upon a clear knowledge of the software applications under consideration. And, you'll want to gain this clear knowledge without investing an inordinate amount of time analyzing the applications. To balance out needs with practicalities, this phase proposes that this research be completed as a series of "passes" in which the number of alternatives is gradually narrowed as more detail is required about each of the alternatives remaining in contention.

Phase II ends with the often-difficult decision of deciding to go ahead with a particular MIS solution. That solution may be a single commercial software application; it may be a combination of commercial software applications that you will bundle together; you may decide to request and pay for modifications in a commercially available package; you may decide to purchase some modules commercially and develop some customized software to supplement the modules and meet your entire information needs; or you may decide to develop custom software for all your information needs. There are so many directions you can go in, and when you arrive at the decision point, you may not be excited about any of the alternatives before you, or you may find that, despite all of your thorough and systematic work, the best alternative is not entirely clear. It may not serve as much consolation to know that you won't be the first to feel this frustration or trepidation.

Following your decision in Phase II comes the work involved in the final phase: System Implementation. Care and diligence during this phase is just as important as in the previous two

[10] The process described here is loosely based on that presented in Chapter 5 of the *CGAP MIS Handbook.* The process has been much more significantly developed and expanded in the current Manual.

phases in order to ultimately achieve success in your efforts. Many of the steps in Phase III can be both costly and time-consuming, depending on the scale of your institution and the complexity of your information needs. In particular, the process of transferring existing data and deciding how to make the transition over to the new system (plunge, pilot, phase or parallel) can be particularly critical decisions.

The process as described in the following chapters will work for a broad variety of microenterprise assistance agencies: small, medium or large institutions; financial-services, non-financial services or multi-service institutions; and non-profit, for-profit or cooperative structures. In some instances, some of the steps can be streamlined, but in no case should any step be skipped entirely. The decision is yours and will need to be made carefully. Some suggestions for when streamlining is feasible are included in the chapters that follow.

MIS Selection Process

Phase I: Defining Needs

- Form a task force
  - Composition of the task force
  - Using a consultant
- Define information needs
  - Collect existing documentation
  - Create process flowcharts
  - Identify where data is collected
  - Identify what people need
  - Assess the current system
  - Project future needs
- Determine what is feasible and appropriate
  - Staff capabilities
  - Cost issues
- Decide on your needs

Phase II: Assessing the Alternatives

- Step 1: Generate and screen list of alternatives
  - Research options (20-30 packages)
  - Screen out clearly inappropriate options
- Step 2: Initial Software Assessment (10-15 options)
  - Group products by module
  - Complete functional comparison tables
  - Assess vendor support
  - Assess each individual module
  - Rate the "fit" for each product
  - Eliminate inappropriate options
- Step 3: Detailed Software Assessment (3-5 options)
  - Thorough testing of demo software
  - Discussions/negotiations with vendors
- Step 4: Final decision and system strategy
  - *Integrated vs. Hybrid?
  - *Use commercial off-the-shelf software, or
  - *Modify commercial software, or
  - *Develop custom software

Phase III:
System Implementation

System Implementation:

- Setting up the hardware
- Preparing and revising documentation
- Configuring the system
- Testing
- Training
- Transfering the data
- Choosing Plunge, Pilot, Phase or Parallel
- System maintenance

3.2. Introduction to the case study

To best demonstrate the MIS selection process, we will follow a single case study throughout this Manual. The case is based closely on the experience of Women's Initiative for Self Employment, a San Francisco-based non-profit. However, in order to demonstrate all the steps in the MIS Selection Process, some of the examples are fictitious. These are identified as such.

Throughout the Manual, the experiences of other institutions will be brought in when appropriate to highlight key issues and demonstrate important experiences.

Case Study: The Women's Initiative Database Enhancement Project

Women's Initiative for Self Employment has operated since 1988, serving low-income microentrepreneurs in the San Francisco Bay area. For the past several years it has maintained three offices in the Bay area, two of which target Spanish-speaking clients primarily in immigrant communities. The Spanish language program is known as ALAS.

In 1993, a database system was started using Paradox™ database software. The system was converted to Microsoft Access™ 2.0 in about 1997 and then to Access™ '97 in 1999.

Because of a convergence of factors — such as the increasing pressure for WI to produce follow-up information for donors, a special FIELD grant to set up an outcome evaluation system that would track historical information about clients, and the unreliability of the existing database system (too many different designers, or "cooks," and different data entry practices over the years) — WI began the Women's Initiative Database Enhancement project in Fall 1999. This process will be documented throughout the remainder of this Manual, along with another fictional case study.

Chapter 4: Phase I: Defining Needs

4.1. Do I really need to go through all of this?

It is nearly impossible to overstate the importance of "starting at the beginning" when it comes to information needs. We are all overworked, with more things to get done than we have the time for. Finding a "solution" for our MIS needs is of urgent importance, and we need it yesterday, not many months from now. The temptation to take shortcuts on the process is great, and you'll probably want to jump right into Phase II and look for solutions. But learn from others that have tried that path — it doesn't work! Taking shortcuts is a major contributing factor to why most institutions are dissatisfied with their information systems. Investing a bit more time and effort up-front can pay off many times over if you can find a better alternative to your needs. If you find yourself under pressure to bypass this phase, remind yourself and others that 100 percent of books about MIS, articles about MIS and MIS experts insist that an institution needs to thoroughly understand its information needs before it can meet those needs.

Still, understanding that the world is far from ideal and that time, financial resources and technical skills are generally in limited supply, we have adapted this needs definition process to be more streamlined than the textbook approach generally advocates. Our working assumption is that most institutions will decide to use mostly commercial software, either as-is ("commercial") or with modest modifications, rather than proceed with the process of developing their own custom software, an expensive and lengthy task.[11] If you expect this is the case for your institution, you can follow the concise approach outlined in this chapter. If you expect to develop custom software or later find that you have no choice but to develop custom software, you will need to perform a significantly more detailed needs assessment, as presented in Section 4.5 of this Manual.

What are the substantial differences between the streamlined approach and the thorough approach to needs assessment? The streamlined approach presented in this chapter focuses on clarification of the flow of information through your institution and your fundamental needs. You will then be able to compare these needs with commercial software on the market to determine if that software provides the information you need in the way that you need it. If things are close but not perfect (and when are things ever perfect?), you'll have enough knowledge to know what modifications to request of the vendor.

The detailed approach laid out in most textbooks walks the users through a systematic approach intended to define and design a system from the ground up, table-by-table and field-by-field. Such an approach is necessary for those developing customized software, but is not necessary for institutions selecting commercial software. In fact, such an approach may even be counterproductive in that after investing such effort in conceptualizing a "perfect" system, the users may be disappointed with perceived flaws in commercial systems and decide that their best alternative is to proceed with the development of custom software when the commercial alternative may have been adequate.

We encourage you to read carefully through this chapter and understand both the process and the purpose of the process. Then, based on your expectations about using commercial

[11] For a more detailed explanation of the "as-is," modified and custom approaches to software, please refer to Chapter 9.

software or developing custom software, you can formulate a decision about the degree of depth for your analysis. But, the question you face is not, "*Should* I do a needs assessment"; the question is, "*How thorough* should it be?"

The steps described in this phase are portrayed in the following figure:

Phase I: Defining Needs

- Form a task force
    - Composition of the task force
    - Using a consultant
- Define information needs
    - Collect existing documentation
    - Create process flowcharts
    - Identify where data is collected
    - Identify what people need
    - Assess the current system
    - Project future needs
- Determine what is feasible and appropriate
    - Staff capabilities
    - Cost issues
    - Decide on your needs

4.2. Form a task force

It is important to insure broad-based participation throughout the needs assessment process and that can best be done by forming a representative task force. The members of the task force can provide guidance and input throughout the process on the critical issues of how the institution works, its philosophy and work culture, and what information people need in order to execute their responsibilities and make sound decisions. Although the level of effort will vary, the task force should typically plan to meet regularly — at least once a week — for a period of perhaps four to six weeks.

4.2.1. Composition of the task force

The task force should be made up of one knowledgeable person from each department.[12] It should include representation from each level in the organization, from senior management to field staff. If you have an information systems department, they should be involved, though not necessarily in charge of the process. In the absence of a true computer technician, you may choose to include a staff person who has advanced knowledge of computers.

An institution with limited in-house expertise may want to hire an external consultant, but this person's role should be clearly defined as one of advising, not decision-making. Because these early stages should not be rushed, it is preferable to hire a locally-available consultant who can devote one or two days a week to the process, rather than work full-time. (See following section for advice on using a consultant.)

[12] This section draws substantially from the *CGAP MIS Handbook,* 66.

A senior person in the organization who has a broad understanding of the institution and commands respect should lead the task force. This is a very important process for the institution, and therefore justifies the close attention and involvement of staff at the highest level. At a minimum, it is helpful in large institutions to have a "project champion" or "sponsor" — an influential person such as the executive director or the board chair, who endorses the process, insures that everyone takes it seriously, and clears bureaucratic hurdles.

4.2.2. Using a consultant

In working with consultants, managers face a challenging dilemma: "How can I evaluate the skills of a consultant and select the best one for my organization when I'm dealing with an area where I have little or no expertise?" And then, "How do I judge the appropriateness of his or her recommendations?"

It is important to first start by clearly defining the project's goals and the consultant's role in achieving those goals. The goals should be specific about the expected outcomes, cost, timeframe and staff resources available to assist in the project. With this list in hand, you can then approach close colleagues, board members and broader-based networks, such as e-mail list serves, to put together a list of referrals that appear qualified and have done acceptable work in the past.

Present your project to several candidates and gauge their response. Do they communicate clearly? Do they tell you what they think you want to hear or are they willing to provide recommendations for real improvement?

Potential questions for the interview include:[13]

- What strengths do you bring to this particular project?
- Have you worked on similar projects or consulted with other groups facing problems similar to ours? What did you learn from the experience? What would you do differently if you could repeat the experience?
- How would you describe the challenges we face from the limited amount you now know about us?
- Describe your work process. How would you work with our staff, board and executive director?
- Which responsibilities must we assume in order to make our work together successful?
- Are you available to complete this work during the time we've specified?
- What else should we know about you, your experience or about what it would be like to work together?

Ask the consultant for a recent — and *complete* — list of clients.[14] Check with a number of past and present clients, selecting clients that appear most similar to your organization, to insure that their work has been satisfactory. Always talk directly with the person who directly supervised the work of the consultant. Discuss the type of work performed, the consultant's strengths and any problems that they might have had with the consultant. Wrap-up the discussion with a question like, "Would you hire this person again?" Some references may be unwilling to talk in much detail; if this is the case, try to get them to at least verify the basic parameters of the work.

[13] Adapted from Verizon Foundation, *Guidelines on Selecting and Hiring a Consultant;* available from http://foundation.verizon.com/06019.shtml; Internet.

[14] Reluctance to provide this information indicates an issue with transparency of the company. See Section 7.4.

Request a bid in writing from the consultant or the firm and make sure that all expectations and outputs are clearly defined in order to minimize future misunderstandings. You should develop a thorough *Terms of Reference* before entering into any contractual relationship with a consultant. **You will find a sample consultant Terms of Reference in Annex 2.** The sample describes all the responsibilities of a consultant assisting the institution through all phases of their MIS selection and implementation. If you anticipate more limited involvement, you would only use those sections of the Terms of Reference relevant to the tasks you expect the consultant to perform. Remember, one of the strengths of this multi-step process is the ability to use "creeping commitment" in your contractual agreements with consultants. For example, hire a new consultant to work with you through Phase I only. In this way, at the end of Phase I, each side can re-evaluate the situation and decide whether they want to continue or end the relationship.

Identify the key staff person that will relate to this consultant and have that person involved in the selection process, making sure that they are comfortable working together.

Finally, remember that you and your organization have responsibilities when working with consultants. Among these are:

- **Honesty:** be forthcoming about the problems your organization faces.
- **Follow-through:** the consultant will rely on your staff to undertake specific activities. Be sure staff members are responsible about completing these tasks in a timely fashion so that the consultant is not held up.
- **Willingness to change:** the consultation may result in recommendations for improving the workflow or operational procedures of the institution. If these recommendations are sound and the changes are important for the remainder of the consultant's work to be implemented, it is your organizations' leadership's responsibility to promote these changes and insure that they are adopted.

4.3. Define information needs

The definition of needs is a critical step. It produces information that will later help to sift through the many alternatives. Handled properly, it can avoid months of frustration and make the difference between success and failure for the entire process.

The steps involved for this process are:

- Collect documentation of existing policies and procedures, i.e., "What is?"
- Create "process" flowcharts.

4.3.1. Collect documentation on existing policies and procedures

The institution needs to first assemble all documentation on its existing policies and procedures. Three main areas require documentation:

- Accounting policies and procedures,
- Basic operating policies and procedures,
- Internal control procedures.

Documentation of policies and practices may be weak — outdated, inconsistent or incomplete — maybe even non-existent. If written documentation is inadequate, staff knowledgeable about how things really work — the unwritten rules — will need to participate actively in this step of the process.

Case Study: Advice on Using a Consultant Well

When Marian Doub started as the Database Coordinator in 1998, Women's Initiative had the good fortune to have two talented computer experts ready and waiting to be of service once the organization was ready for them. This life- and business-partner team of a database systems analyst (or database developer) and a systems engineer had chosen to make a sizable contribution of their time to Women's Initiative as volunteers. However, until the moment of Marian's hire, there had been little staff time available to coordinate use of their expertise. After they assisted with her orientation, it took her eight months to prepare: to get the gist of the organizational needs as well as complete a system upgrade (new computers, network, and go from Access™ 2.0 to Access™ '97 in preparation for Year 2000 [Y2K] and beyond). They advised along the way during this preparation phase, but did not become fully involved until a year after Marian had started her job.

The systems analyst coordinated all other volunteers and interns who contributed and she did quality control on all pieces and aspects. She reviewed all changes and enhancements before incorporating them in the system. The three worked closely together over the next three years. To date, Women's Initiative estimates having received an in-kind donation of approximately 2,800 hours from the consultant team! During this process, the team learned the following lessons, equally applicable with volunteer or paid consultants:

Be ready. Have a high degree of organizational readiness:

- Up-to-date computer technology (hardware, software, network[s]);
- Enough staff capacity (time and knowledge) to interact and manage the project with the consultant;
- Clarity and finality about what data needs to be managed (at least a collection of all the hard copy data collection forms and reports used by the organization, as well as a clear explanation of how it all works together); and a
- Back-up plan if the consultant does not work out or if the project takes longer or takes more resources than expected. If possible, get a sanity check from a third party before committing to the project: is it realizable and cost effective?
- Acceptance that these items change over time: components become outdated and are replaced, staff turnover happens, data needs are constantly being refined.

Be available and responsive to the consultant's need for information about organizational needs and assumptions. Constant communication is a must for a successful outcome.

- Staff person dedicated to working closely with the consultant for the duration of the project, to advise and provide information from the "user" side.
- Set up clear and consistent meetings with the consultant to monitor progress and manage priorities.
- Constantly test and question assumptions as well as products.
- Emphasize prioritizing, so that key components get done first.

If written documentation is incomplete, it is not necessary at this time to generate or revise things. Since there is the likelihood that procedures will need to be refined and updated when the new MIS is in place, it is best to postpone document revision until the transition to the new system is underway.

4.3.2. Create process flowcharts

The documentation on policies and procedures assembled serves to diagram the flow of information through the institution. We will introduce flowcharts with two levels of detail. The goal of these flowcharts is to discover answers to these questions:[15]

- Where are data collected?
- Where are data transformed into information?
- Who needs what information?
- What decisions need to be made?
- What information is required to make those decisions?
- When do the decision makers need it?
- Where is information stored?
- Where can re-engineering make processes more efficient?[16]
- Where are the leverage points and critical processing points where a change in procedure could significantly improve efficiency and service?

Using process flowcharts to understand your institution

Surprisingly, many institutions and many staff within institutions do not always have a clear understanding of all the steps related to a client receiving services and how each individual step fits into the overall process. Even in organizations that have a more systematic understanding of their "production process," perceptions are generally clear around the basics of the "best case" scenario, but considerable ambiguity exists where anomalies or "exceptions to the rule" take place. Although manual systems can tolerate ambiguity, the more automated the system, the more precisely the system needs to be designed and the more complex it becomes.

MIS consultants frequently request any existing process flowcharts early in their diagnostic phase. They then use these during their staff interviews to clarify their understanding of how everyone's work ties together and how information flows throughout the institution. During such a process, two things tend to happen.

Firstly, these flowcharts generally go through a number of revisions as everyone becomes clearer about the process and their role in it. In particular the "exceptions" get noted, for example, where supervisors can approve marginal cases, or what the procedure is for a client who gets approved for a loan, doesn't take it for some reason and then comes back later requesting the loan be disbursed. Drawing a detailed flowchart, complete with diamond-shaped decision boxes at each point where decisions can be made, forces users to consider each and every situation — not just the typical path. As already mentioned, these exceptions are not trivial when dealing with software.

[15] This list originally appeared in the *CGAP MIS Handbook,* 67.

[16] Reengineering has become synonymous with "downsizing" and laying-off staff, particularly middle-management staff. However, the original concept presented by Michael Hammer and James Champy in their book *Reengineering the Corporation (Harper Business, 1993)* was to rethink processes and procedures in order to streamline them and delegate more decisionmaking to front-line staff in order to enhance service and reduce response time. Technology was proposed as a critical tool in achieving these goals.

Secondly, once the flowcharts are complete, staff and management have a clear and complete picture of the process, often for the first time. And, as that process is scrutinized, it often becomes clear that there are many opportunities to improve the process. Management are then faced with a crucial dilemma: to look for a means to automate their existing systems, and thus, lock in these inefficiencies, or to first revise their procedures, undertaking a type of "re-engineering" process, and then to automate. The second alternative is what often results in much more extensive and time-consuming efforts than initially anticipated, but it also often results in new and creative ways to process work in the organization.

Managers who have encountered this situation often say, "That's not what we expected when we started out on this process, and if we had known, we actually may not have ventured forward. But, now that we have committed to this work and the effort is mostly behind us, we're glad we did it. We're a better institution as a result of what we've gone through."

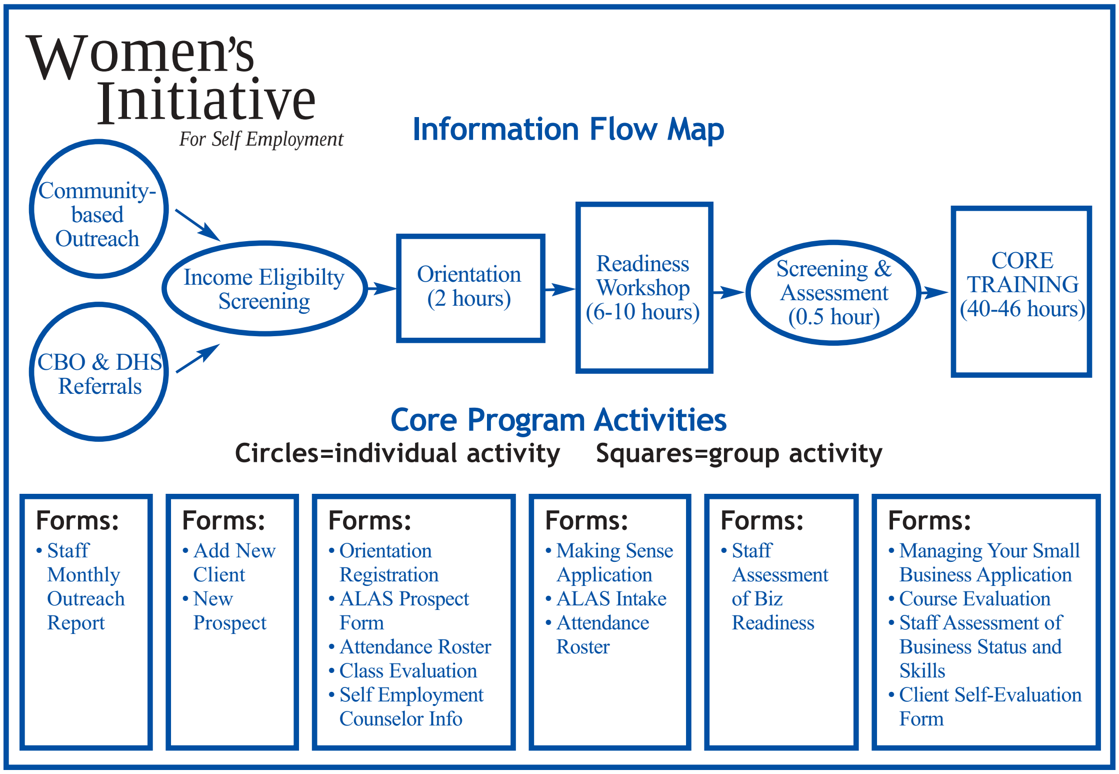
Producing process overview flowcharts

The first level of flowchart to produce is an "overview" chart that depicts in broad strokes the overall services available to the client. This is particularly important in institutions providing a range of services. Senior managers, who understand the overall process and how the different products and departments of the institution work together, generally can provide the necessary information to prepare overview charts.

Case Study: Process Overview Flowcharts

The consultant and staff worked together to develop two information flow maps documenting the processes in place at Women's Initiative. The first depicts the initial client outreach and screening followed by core training activities. Note that the map (on the next page) combines an overall flow in the top half of the diagram with a listing of the related information-gathering forms on the bottom half.

The second information map then depicts the range of other services available to clients who have completed the core training.



Women's Initiative

For Self Employment

Information Flow Map

Community-based Outreach

CBO & DHS Referrals

Income Eligibilty Screening

Orientation (2 hours)

Readiness Workshop (6-10 hours)

Screening & Assessment (0.5 hour)

CORE TRAINING (40-46 hours)

Core Program Activities

Circles=individual activity Squares=group activity

Forms:
- Staff Monthly Outreach Report

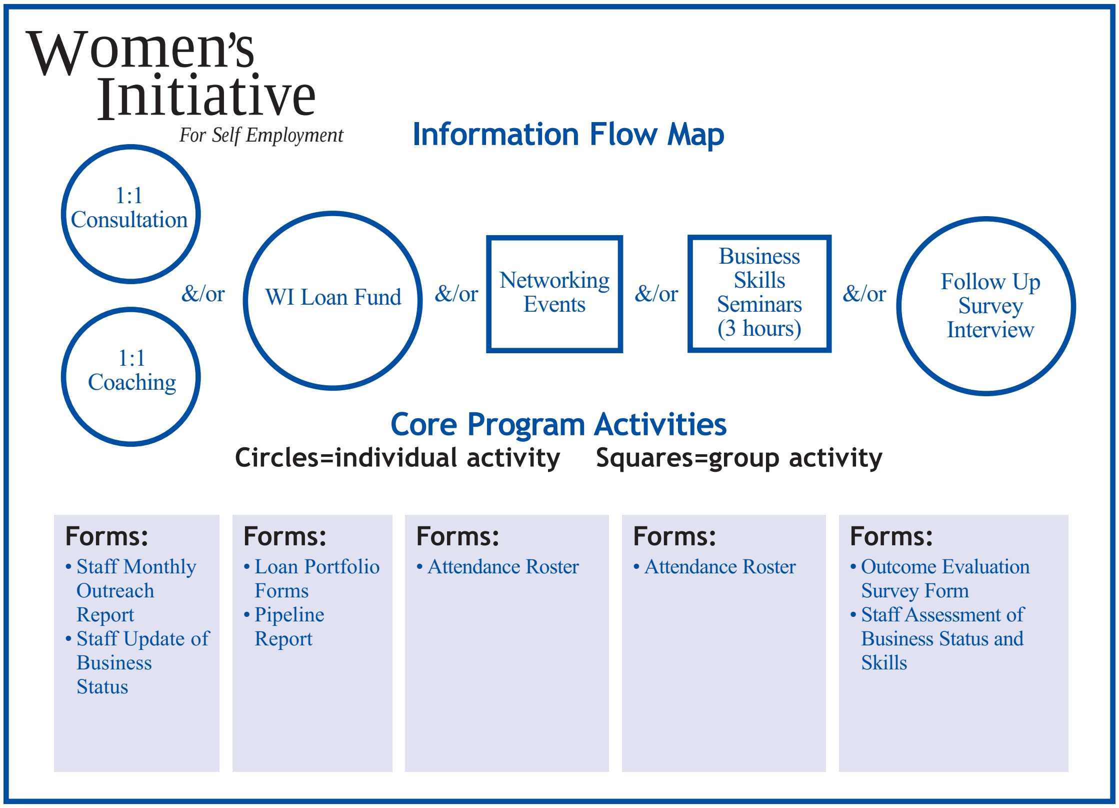
Forms:
- Add New Client
- New Prospect

Forms:
- Orientation Registration
- ALAS Prospect Form
- Attendance Roster
- Class Evaluation
- Self Employment Counselor Info

Forms:
- Making Sense Application
- ALAS Intake
- Attendance Roster

Forms:
- Staff Assessment of Biz Readiness

Forms:
- Managing Your Small Business Application
- Course Evaluation
- Staff Assessment of Business Status and Skills
- Client Self-Evaluation Form



Women's Initiative

For Self Employment

Information Flow Map

1:1 Consultation

1:1 Coaching

&/or

WI Loan Fund

&/or

Networking Events

&/or

Business Skills Seminars (3 hours)

&/or

Follow Up Survey Interview

Core Program Activities

Circles=individual activity Squares=group activity

Forms:
- Staff Monthly Outreach Report
- Staff Update of Business Status

Forms:
- Loan Portfolio Forms
- Pipeline Report

Forms:
- Attendance Roster

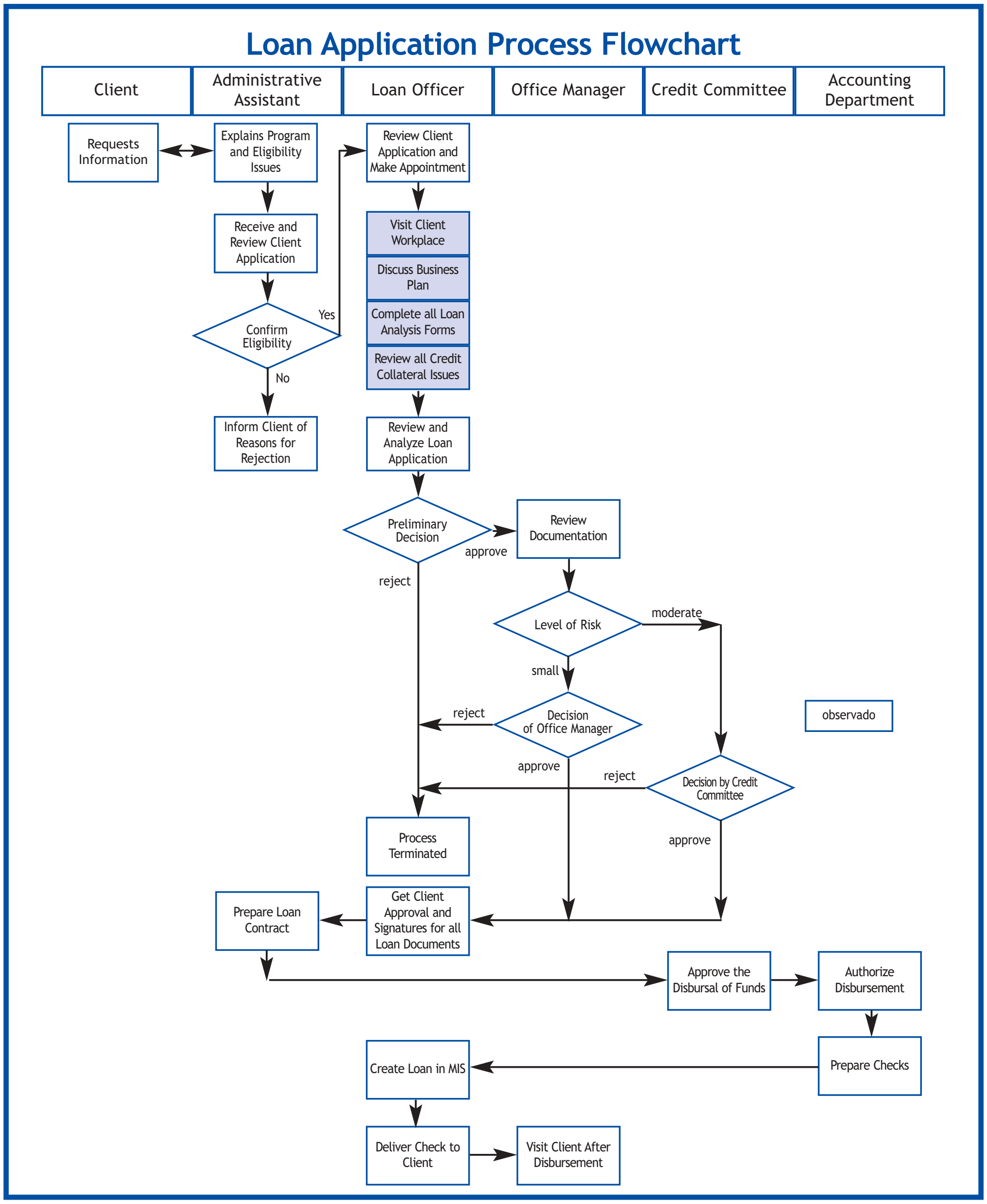
Forms:
- Attendance Roster

Forms:
- Outcome Evaluation Survey Form
- Staff Assessment of Business Status and Skills

Producing detailed process flowcharts

Each of the services provided and noted in the "overview" flowchart can be further described in a more detailed flowchart. For example, all steps involved in applying for a loan can be described in a dedicated flowchart. A useful format for this more thorough flowchart is to identify columns for each key staff person or department, including the client (see example in figure below).

Such charts will likely need to be developed through an extensive interview process with all staff. The interview should ask users what they do and why they do it. What information do they need? What do they do with it? Why? To whom do they send information and reports? Why? It is important to discover how users perceive their place in the process. The analyst can sketch out flows during the interview process in order to check their accuracy. They should review for accuracy again after a cleaned-up draft of a process has been prepared.


Loan Application Process Flowchart
Client
Administrative Assistant
Loan Officer
Office Manager
Credit Committee
Accounting Department
Requests Information
Explains Program and Eligibility Issues
Review Client Application and Make Appointment
Receive and Review Client Application
Visit Client Workplace
Discuss Business Plan
Complete all Loan Analysis Forms
Review all Credit Collateral Issues
Yes
Confirm Eligibility
No
Inform Client of Reasons for Rejection
Review and Analyze Loan Application
Preliminary Decision
approve
Review Documentation
reject
Level of Risk
moderate
small
reject
Decision of Office Manager
observado
approve
reject
Decision by Credit Committee
approve
Process Terminated
Prepare Loan Contract
Get Client Approval and Signatures for all Loan Documents
Approve the Disbursal of Funds
Authorize Disbursement
Prepare Checks
Create Loan in MIS
Deliver Check to Client
Visit Client After Disbursement

In complex institutions, a separate flowchart should be created for each major process. For example, in a lending institution, you might prepare flowcharts for: first-time applications, post-disbursement procedures, follow-up loans, refinancing procedures, legal actions against delinquent clients, etc.

An example of a detailed process flowchart is provided at right.[17]

4.3.3. Identify where data is collected

Reviewing all paper-based forms is the most appropriate and straightforward place to begin this step. In most institutions, data is first written on paper and then selected data is transferred to MIS software. Collect a copy of every form in use and organize them by the

[17] Note that this sample does not pertain to the Women's Initiative case study.

steps in the process, as was done in the previous Women's Initiative sample overview flowcharts.

A typical institution may have such forms as:

- New Prospect Form,
- New Client Form,
- Course Registration Form,
- Course Attendance Form,
- Loan Application Form,
- Loan Approval Form,
- Staff Assessment of Business Skills, and
- Loan Payment Receipt.

It is important to collect actual copies of each of these forms, preferably filled in with data, to clarify the way in which the information is collected.

4.3.4. Identify what information people need for decisionmaking

Using Indicators to Track Performance

The reason we need to collect so much data in our institutions is so that staff at all levels will have meaningful information on which to base decisions. Some data is useful in "raw form," such as contact information, loan balances, course attendance, etc. However, there is a wide variety of areas within the institution that are best monitored by use of performance indicators.

It is critical that appropriate indicators be identified and carefully defined during the needs identification process, as the MIS ultimately implemented must be capable of tracking the necessary data by the required definitions, from appropriate data sources (e.g., forms) and within desired time intervals. This is definitely *not* a straightforward process. Rather, it is one of the more challenging areas for MIS for a variety of reasons, some of which were explained in Section 2.3 on the dimensions of complexity with databases. Many managers have been surprised and even angered to find out that after months of collecting data, they are not able to use that data to extract the indicators they need to do their job — monitor performance, track progress, report to stakeholders and plan for the future of the institution.

There is no complete consensus on all indicators that a microenterprise assistance institution needs to track. However, through efforts such as FIELD's MicroTest project and the CDFI Data Project, there has been significant progress in the past several years. These efforts are described in the following section.

MicroTest Framework

MicroTest was initiated in 1997 by and for microenterprise development practitioners[18] interested in documenting and improving upon the level of performance in the microenterprise field. Its mission is to improve the quality of microenterprise services and

[18] The original "Core Group" of MicroTest programs included: ACCION US; Detroit Entrepreneurship Institute; FINCA USA; Institute for Social and Economic Development; Maine Centers for Women, Work and Community; Northeast Entrepreneur Fund; PPEP/PMHDC/MICRO; Rural Economic Development Center; Self-Employment Loan Fund; Western Massachusetts Enterprise Fund; Women's Economic Self-Sufficiency Team; Women Entrepreneurs of Baltimore; Women's Initiative for Self Employment; Worker Ownership Resource Center; and Working Capital.

the stability of microenterprise organizations over time by perfecting and promoting the use of measures to regularly assess performance.

From an initial cohort of 13 programs in 1997, MicroTest has grown to 56 in 2002. The MicroTest membership has grown in a variety of ways in a few different cycles. The first two groups of members joined MicroTest on a voluntary basis. The membership process was competitive and emphasized each program's: 1) organizational track record; 2) interest in performance measures; 3) motivation to participate in a learning effort with industry-wide implications; and 4) in-house information gathering capacity.

The primary task of the original cohort was to develop, define and test a set of performance measures for the field. It took more than a year of facilitated discussion, testing and consensus-building for all MicroTest performance measures to be finalized. These measures are presented and defined in detail in Annex 1. What follows below is the analytic framework for organizing and interpreting these measures. Depending on your organization's mission and goals, these and other performance-related questions will be important for you to consider as you assess different MIS design options.

MicroTest Performance Framework

MicroTest focuses on categories of outputs by which microenterprise development providers can assess their performance. While each category is not appropriate for every program, this framework captures the "pillars" of microenterprise performance. It is designed to capture a range of program performance areas (not client- or borrower-level outcomes data) and to mitigate the tendency to evaluate complex program(s) according to one-dimensional criteria. What follows is an overview of the key questions MicroTest performance measures address.

Reaching Target Groups

- Who is the program actually serving?
- Is the program fulfilling its outreach mission?

Achieving Program Scale

- How many clients received credit and/or training-related services?
- What is the magnitude of program services delivered in a fiscal year?
- What is the volume of lending activity?

Credit Program Effectiveness

- What is the quality of the portfolio?
- How does the level of risk in the portfolio influence portfolio quality?

Training Program Effectiveness

- To what extent does the program succeed in assisting clients to achieve key training objectives?

Program Efficiency and Sustainability Measures

- How efficiently does the program use internal resources?
- How self-sufficient is the program?
- How diversified is its funding?

MicroTest members continue to report an increased capacity to use MicroTest data as a decisionmaking tool to improve their programs. Current MicroTest directions include testing a key set of simplified client-level outcomes questions; interested programs are encouraged to contact FIELD staff for more information about a simple “outcomes monitoring” toolkit. For further information about MicroTest, contact FIELD staff and look for a forthcoming statistical review of all MicroTest data in the summer of 2002.

4.3.5. Assess the current system

The task force should include an analysis of the current system, even if the intent is to completely replace it.[19] It will be valuable to identify both the current system’s strengths and weaknesses in order to pinpoint issues that should be addressed by the new system.

The following questions can guide this review:

- What type of system is it — manual, computerized or a combination?
- What different modules exist in the system and how are they linked together?
- What skills are required to use and maintain the system?
- What are the system’s strengths and weaknesses?
- Can the system be expanded or improved?
- How satisfied are the system’s users?
- What are the causes of dissatisfaction?

4.3.6. Project future needs

Predicting future needs is a critical part of the task force’s job.[20] A MIS should be expected to have a minimum life of five years, adapting to the institution’s changing needs as it grows. To project those needs, it is helpful to consult the institution’s strategic plan. These are some of the questions that should guide the discussion of future needs:

- What rate of growth is expected?
- What reorganizations are expected?
- What changes in workflows are expected?

Case Study: Assessment of Current System

In 1993, a database system was started using Paradox™ database software. The system was converted to Microsoft Access™ 2.0 about 1997 and then to Access™ ’97 in 1999.

The system never was able to support synchronization of data among the three branch offices, so all data entry and retrieval had to be performed from the San Francisco central office.

Data storage was spread among four different Access™ databases, of which each needed to be opened independently. In the past, many different staff had the ability to both read and update (including deleting from) the databases, so (as is typical of that situation) changes to the data were not always made consistently and some data had been lost. In 1999, all access to the databases was restricted to three people. The Data Coordinator and the Workshop Coordinator perform the bulk of data entry. The Data Coordinator is now ultimately responsible for the completeness and accuracy of the data that has been entered as well as the reports.

[19] This list originally appeared in the *CGAP MIS Handbook,* p. 69.

[20] This section is based on the *CGAP MIS Handbook,* 70-71.

- What changes in *existing* products and services are expected?
- What *new* products and services are expected?
- What issues of centralization and decentralization are expected?
- Are new funders or stakeholders anticipated that may affect reporting requirements?
- What new areas of information needs, e.g., outcomes reporting, are anticipated?

Answering these questions precisely may be difficult, but the key issue lies in just two questions: "Is the institution expected to be stable (with the exception of growth)?" and, "Does the institution have a culture of innovation, always modifying products and procedures and trying new things, or does it tend to stay with what works best?" An institution that frequently changes procedures and introduces new products is better off with an option that includes establishing the in-house capability to modify and adapt its system.

4.4. Determine what is feasible and appropriate

Once the institution's needs have been defined, it is time to assess what is feasible. A principal job for the task force is to determine what degree of computerization is feasible by assessing staff capabilities and cost considerations.

4.4.1. Staff capabilities

The capability of staff to manage computers is critical to the successful incorporation of new computer technologies. The task force needs to examine the following issues:

- Who will be managing the new system? Is there an internal information systems department or will the institution rely on contracting outside support?
- If local consultants or an outside company will provide ongoing support, are they competent, reliable and affordable?
- How much training of users will be required? Will this training be provided in-house or by an external source?[21]
- How strong are the accounting department staff members? Will they be able to handle a sophisticated system? Are they able to keep information up to date?
- What level of complexity in computer systems can be supported at the head office and at any branch offices?

4.4.2. Cost issues

When budgeting for a system or comparing the prices of systems, it is important to look at the total cost before making a decision. The cost of the software may be inconsequential compared to the cost of staff time associated with configuration of the new system, data transfer and staff training, or the cost of upgrading all computer hardware or the operating system required to run the new software. The cost of annual support or updates often matches or exceeds the original cost of the software.

"What does a MIS cost?" is a difficult question to answer. As described in the CGAP Handbook on MIS,[22] it is not unlike trying to answer the same question for someone shopping for a new car. In both cases, the answer depends on many factors:

[21] Don't underestimate the amount of training that will be required for any new systems! See Section 10.7 for further information on staff training.

[22] See *CGAP MIS Handbook,* 73.

- **What will the purchase be used for?** Short trips to the grocery store, essential transportation to work or racing in the Grand Prix? Each of these answers will result in a different purchasing decision.
- **How large is the family?** An institution that has to carry a lot of passengers will need a vehicle that accommodates them.
- **How long does the purchase need to last?** A long-term decision will mean a different choice than an interim purchase.
- **How much is the institution able and willing to spend?** Prices vary tremendously. Purchasers often narrow their range of options by considering only what they can afford. Perhaps the budget allows shopping only for used cars or for low-end rather than luxury cars.
- **How important are the options?** "Bells and whistles," like digital stereos and cruise control, raise the cost above the basic model or may not even be available on the model under consideration.

Clearly, situations, needs and available resources vary too widely to give precise guidelines on how much to budget. But, in preparing a budget, an institution needs to be sure to consider a wide range of potential expenses.

The following worksheet itemizes these categories and provides space to estimate costs. Three columns are provided: best-case scenario cost (lowest cost), most likely cost and worst-case scenario cost (highest cost).

MIS Cost Estimation Worksheet

Category	Best-Case	Most Likely	Worst-Case
One-Time Expenditures			
Software Costs			
Purchase price			
Customization			
Licensing of other software (operating system, database engine, etc.)			
Configuration			
Data Transfer support (technical assistance, data entry staff)			
Training expenses (trainers, travel, staff overtime, etc.)			
Hardware Expenses			
Servers			
Computers			

MIS Cost Estimation Worksheet			
Category	Best-Case	Most Likely	Worst-Case
Printers			
Back-up power supplies			
Data back-up system			
Infrastucture Improvements			
Wiring			
Workspace modifications			
Other			
Total One-Time Expenditures			
Annual Recurring Costs			
Insurance premiums			
Annual software maintenance			
Annual software licensing			
Annual hardware maintenance contract			
MIS Department staffing			
Annual licensing of other software (operating system, database engine, etc.)			
Total Annual Recurring Costs			

All these costs will depend on choices of head-office and branch-office computerization and between front-office and back-office operations, accounting and portfolio system computerization, and custom and off-the-shelf software.

Several FIELD MicroTest members provided information on the costs of their MIS — Northeast Entrepreneur Fund, North Star CDC, West Company and WESST Corp. Their responses were used to generate the sample cost data below.

Category	Typical cost ranges
One-Time Expenditures	
Software costs	
Purchase price	Costs started at $1,000 per module. Multiple module systems were $2,000 to $3,000 with one system costing $17,000.
Customization	All respondents had systems requiring either customization of a commercial module or development of a custom module. Costs ranged from $1,000 to $8,500.

Licensing of other software (operating system, database engine, etc.)	This was dependent upon the system purchased. Only one respondent incurred costs here of $2,000.
Configuration	Configuration was generally included on the cost of the software. However, one respondent paid $2,500.
Data Transfer support (technical assistance, data entry staff)	Two respondents needed assistance with data transfer at costs of $1,000 and $2,500.
Training expenses (trainers, travel, staff, overtime, etc.)	Only one respondent had costs here of $500.
Hardware expenses	Costs varied widely here, from $8,000 to $43,000 to $59,000.
Infrastructure Improvements	Costs were minimal in this category except for wiring expenses of $200 for one and $2,500 for another.
Total One-Time Expenditures	Total one-time expenditures ranged from $13,000 to $48,000 to $83,000.
Total Annual Recurring Costs	Annually recurring expenditures were quite minimal with totals of $1,000, $1,200 and $2,400. None of the respondents had expenses in the potentially largest category - MIS staff.

As demonstrated by the survey data, costs of installing a MIS are considerable. Remember, however, that although the cost of installing a MIS may be high, the cost of not having information is higher. For most institutions, the benefits of investing in information can quickly exceed the costs of moderately-priced systems. Often, the best strategy is to invest for the long term. Paying more now for a system that will serve the institution longer can mean lower annual costs over the life of the system.

4.5. Decide on your needs

There are two steps remaining to complete the needs assessment. The first is to review current procedures and determine if any changes and improvements are necessary. The second is to generate a basic ranking of needs that will serve as a screening tool when moving forward into the next phase of the MIS selection process.

4.5.1. Make any necessary adjustments in procedures

As already mentioned frequently, a thorough and systematic introspective process may uncover a list of ideas on how to improve, rationalize and streamline the institution's policies and procedures. Tasks may currently be redundant, responsibilities may not be properly divided among staff, and procedures may result in lengthy delays and irritations for clients. This is the opportunity to take documentation reflecting "What Is" and modify to reflect "What Should Be." This is particularly critical as the MIS decision to be made shortly must be suitable for the institution as it expects to be in the future.

This is a process that can either be brief, if the institution limits itself to a few minor but effective changes, or it may take considerable time and effort, if the institution decides to completely revise its procedures. This is a critical decision for the institution and should be made with care.

In some cases, the future MIS selection can play a critical role in re-engineering the way in which the institution works, as the institution chooses technology precisely because it does change the entire workflow. The example of ACCION Texas illustrates this point.

Case Study: ACCION Texas and electronic networking

In 1998, ACCION Texas developed electronic networking technology which enables loan officers to process loans from any PC with an Internet connection. Staff members are no longer limited to being in the office to access their information, but have 100 percent of their files and applications available in real time from anywhere they find themselves, even a client's business or their own home.

The system provides complete tracking of an application from initial contact to closing to loan collection and charge-off. Before the use of this system, loan officers completed forms which they faxed into the main office where a staff person then entered data from the forms into the computer system. A printout was then faxed back to the loan officer for review. This back-and-forth exchange continued through the loan processing phase. Data entry was often duplicated and it was cumbersome to change or correct information when out in the field. Repetitive processes resulted in increased costs and delays in loan processing.

ACCION Texas believes their new loan processing procedures, supported by their ClientTrack™ technology, have enabled them to significantly lower their cost per loan while simultaneously improving service to clients. They believe they have already recouped their investment in MIS many times over.

4.5.2. Generate a basic ranking of needs

The final step in Phase I is to provide a basic ranking of your institution's needs using the frameworks presented in Section 2.3 — the three challenges of complexity for MIS — and an approximation of level of complexity for each of the modules you are considering.

The figure on page 54 provides a worksheet to record this information. Data has been filled in for the Women's Initiative case study. Note that blank copies of all worksheets included in this Manual can be found in Annex 4.

Ratings for Your Institution

Category	Basic (1 point)	Intermediate (2 points)	Advanced (3 points)	Score
Breadth		X		2
Depth		X		3
Scale		X		3
Total Score				8

Modules	Rating (Basic, Intermediate, Advanced)	Comments
Loan Portfolio	2	Currently used module is acceptable.
Demographics demographic	3	We need to track detailed information for comparison with outcomes data.
Contact Mgmt.	2	
One-on-One TA	3	
Training Courses	2	
Outcomes	3	We have an extremley strong commitment to tracking and analyzing outcomes.
Accounting	2	Our current module is acceptable.

Case Study: Summary of the Women's Initiative Needs-Assessment Phase

The database developer was the project manager with my supervision and participation as a member of the staff Database Task Force. During the first two months of the project, she conducted the staff needs assessment and instructed them about good database design. The needs-assessment report and presentation guides have been useful ever since.

Although the framework presented here did not exist at the time we did our assessment, we would have assigned the following scores to each of the categories:

Women's Initiative provides training classes, credit and individual technical assistance, scoring a "2" in breadth of services. We are very committed to monitoring and measuring the outcomes of services, and thus have chosen to collect a significant amount of information on clients and services, scoring a "3" in depth of information. We operate two programs, one in English and one in Spanish. The two programs have some differences in operating procedures and are run from three different offices. Women's Initiative also has achieved significant scale, with more than 9,600 clients and at least 900 businesses having shown business growth, and currently has 26 staff members. Thus, we score a "3" in scale, for a total score of eight out of a potential nine.

4.6. Conclusion

This first phase in the MIS selection process has covered a lot of ground, looking at the institution as it is, envisioning where it is heading and thinking of ways to make it work even better. The process has involved staff in identifying strengths and weaknesses, and articulating the realities of what type of MIS is most suitable.

If done properly, the needs assessment will form an invaluable foundation for proceeding into the next phase of the selection process: Assessing the Alternatives.

Chapter 5: Phase II: Assessing the Alternatives

The challenge in Phase II is to determine the best course of action for meeting the institution's MIS challenges. This requires identifying the alternatives available to the institution, assessing these alternatives in light of its specific and unique needs as determined in Phase I, and then determining a strategy for moving forward. The task can be difficult because, except in rare instances, the alternatives available fall short of ideal and the best choice is often far from obvious. The alternative, to not clearly assess the alternatives, is worse — akin to playing "pin the tail on the donkey."

The process described here can be quite intimidating. It looks both complex and tedious. In part, this is due to the need to write a single manual that meets the needs of a wide range of institutions. As you look at the structure of this and the following chapters you will recognize entire sections that you can pass over based on the needs of your institution. In addition, many of the techniques described for use in evaluating software modules have already been applied to the range of software products included in the companion document to this Manual: ***2002 FIELD MIS Software Review.*** This Manual describes the techniques so that you may both understand the issues in more depth as well as be able to apply the techniques to evaluate any new software products not included in the FIELD Review.

5.1. Overview of the steps in Phase II

The following figure describes the four steps to be followed in Phase II. The overall purpose of the four steps is to carefully assess the fit between the institution's practices, both present and future, and the capabilities of the various software packages under consideration. As the fit is virtually never perfect, the selection process needs to begin with clarity about the functionality the institution expects from the MIS and the degree to which it is willing to adjust its procedures to match the MIS.

Phase II:
Assessing the Alternatives

Step 1: Generate and screen list of alternatives
- Research options (20-30 packages)
- Screen out clearly inappropriate options

Step 2: Initial Software Assessment (10-15 options)
- Group products by module
- Complete functional comparison tables
- Assess vendor support
- Assess each individual module
- Rate the "fit" for each product
- Eliminate inappropriate options

Step 3: Detailed Software Assessment (3-5 options)
- Thorough testing of demo software
- Discussions/negotiations with vendors

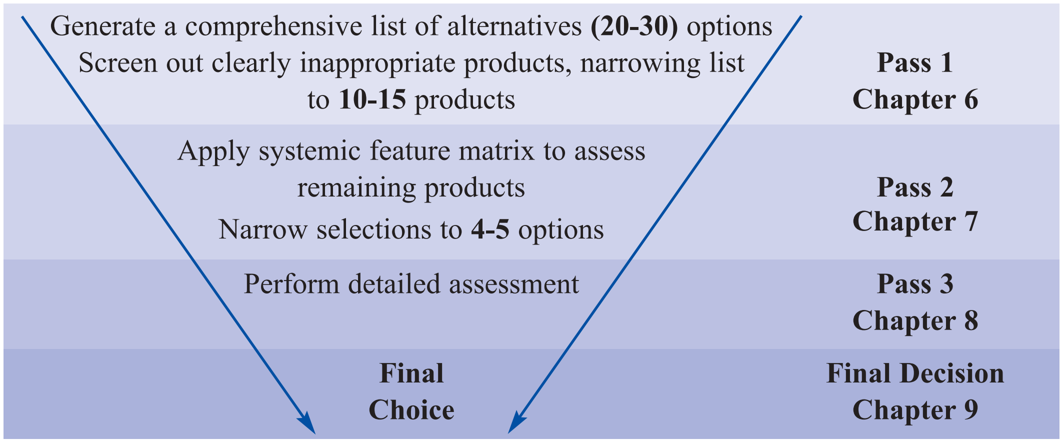
Step 4: Final decision and system strategy
- *Integrated vs. Hybrid?
- *Use commercial off-the-shelf software, or
- *Modify commercial software, or
- *Develop custom software

All institutions should always start by evaluating existing commercial software options. Even in the rare case where an institution expects to ultimately develop its own software, a sound starting point is to study other software and learn what approaches they take to data collection and storage.

In evaluating software, it is often easier to eliminate a program from consideration than to determine whether it will be fully compatible with the institution's needs. The assessment should therefore be divided into multiple passes to narrow down the alternatives at the same time that the level of detail in the analysis increases.

The intent in Phase II is to research a list of possibilities and begin to screen the list down to a manageable size with a realistic amount of effort. It is important to not stop short on this initial list. There may be an ideal solution for your institution that never even gets consideration! This chapter describes approaches and techniques that you can use to both generate the list of possibilities and perform this initial screening.

Overview of Process to Evaluate Commercial Software Options



Generate a comprehensive list of alternatives **(20-30)** options
Screen out clearly inappropriate products, narrowing list to **10-15** products

Pass 1
Chapter 6

Apply systemic feature matrix to assess remaining products
Narrow selections to **4-5** options

Pass 2
Chapter 7

Perform detailed assessment

Pass 3
Chapter 8

Final Choice

Final Decision
Chapter 9

The process is comprised of three "passes" through the alternatives. The first pass is to assemble the list of possibilities, perhaps 20 to 30 options, by drawing information from a broad range of sources, and then to quickly eliminate clearly inappropriate options. The second pass is the "initial" software assessment to look at main features of the 10 to 15 remaining products to determine which are the most likely final contenders. The final pass is then to perform a "detailed" software assessment of the three to five top candidates and reach one of the following decisions: (1) a single package that meets your needs with or without modification, (2) two or more packages that you intend to blend together into a hybrid system, or (3) none of the alternatives is suitable and you need to consider developing custom software. The alternatives and their implications are described systematically in Section 9.8, "MIS Alternatives Decision Tree."

This process can be visualized as gradually sifting through the commercial products on the market, first learning "a little bit about a lot of products," gradually eliminating inappropriate products and working toward learning "a lot about a few products." Ultimately, the choice ideally comes down to a single product. This process needs to insure that you have the necessary information to make the right decision.

It must be stressed again that the goal here is not to identify the most widely-accepted product. That would be relatively easy. Instead, MIS software is an intensely individualized product. Software that one institution raves about can very well be totally inappropriate for another institution. Equally important, one institution having a "bad experience" with a product does not necessarily mean that another institution should rule out the product. The unfortunate experience of the first institution may well have been the result of a poor choice of software. The purpose of a systematic process such as that presented here is entirely to minimize the chance that your institution makes a poor choice. Don't take shortcuts on the process! A bit of extra effort expended early on can eliminate a tremendous amount of frustration later.

Chapter 6: Phase II, Step 1: Generate and Screen Alternatives

6.1. Research software options to generate a comprehensive list

Our task begins by assembling a list of the commercial products that are currently available. Typically, expect your initial list to contain about 20 to 30 alternatives. There is a variety of means available for putting together this list:

- **Review existing "overview" documents**

You aren't the first person who has had to decide which software to buy. Others have gone through this same process and some have made their work publicly available.

As explained in the **Background** section of this Manual, the first publication of the FIELD MIS Study was a **Findings Report,** which provides annotated reviews of 10 software packages. This study has been updated and is available as a companion volume to this MIS Manual, the ***2002 FIELD MIS Software Review.*** As the software field is constantly evolving, check at www.fieldus.org to see if any updated information has been made available.

The National Association of Development Organizations (NADO) has published a study entitled ***Loan Servicing Software****,* providing information about 12 programs. The study includes contact information, key features, pricing and service. The updated second edition was published in June 2001, is 10 pages in length and costs $10. It is available at: www.nado.org.

The National Community Capital Association (NCCA) has a similar study, which in addition to describing the software selection and implementation process, provides a basic overview of widely used products and vendor contact information. It was published in 1998 and is available at: http://www.communitycapital.org/resources/operations.html.

- **Attend software vendor fairs**

Software vendors often participate at major events held for microenterprise practitioners, such as the NCCA Annual Training Conference and the AEO Annual Conference, in an effort to interact with current and future clients. Take advantage of these opportunities to collect information from the vendors and get a preview of the look-and-feel of the software products.

- **Poll colleagues**

Use the broader network of microenterprise practitioners to assemble leads and opinions on various products. Call or e-mail around to colleagues. Use existing Internet "list-serves" dedicated to microenterprise issues, such as "CommDev," managed by Cornell's Community and Rural Development Institute, or "MicrofinancePractice," managed by Chuck Waterfield and Howard Brady.[23]

- **Web research**

Virtually every commercial software product has a Web site that provides additional information and, many times, downloadable demo versions.

[23] To subscribe to "CommDev," visit www.alternatives.org/cdblist.htm. To subscribe to "MicrofinancePractice," send a blank e-mail to MicrofinancePractice-subscribe@yahoogroups.com.

6.2. What you are likely to find

Your research is likely to have varying degrees of success, depending on the modules you need and the degree of integration of modules you expect. (For background on these concepts, please refer to Section 2.3.1.)

Firstly, expect to uncover a large number of **loan portfolio modules** that are suitable for the U.S. microenterprise market. The number of commercial offerings is due to several factors:

1. The market for loan portfolio software is broader than specifically microfinance. There is a large market among CDFIs, cooperatives and credit unions, development finance institutions and finance companies.

2. Loan portfolio management in the United States generally follows consistent rules in the calculation of interest, repayment schedules and aging.

3. Institutions give high priority to managing their portfolios because they are fiscally accountable for the funds.

4. Demand has also increased for loan portfolio software due to the need to accurately and efficiently calculate interest and penalties owed.

Unfortunately, non-financial modules, including **Demographic modules** and **Contact management modules,** are far less common on the commercial market. This is in part due to the following reasons:

1. There is a wide diversity of services, and modules must be developed or adapted for each service.

2. Delivery methodologies vary among agencies, complicating the development of generalized software.

3. There is insufficient agreement on definitions and indicators for non-financial services, inhibiting the development of commercial solutions.

4. There is a smaller market for non-financial modules. Agencies provide fewer non-financial services than loans.[24]

5. The price that agencies are willing and able to pay for software modules may not be high enough to attract many commercial offerings.

6. Commercial software companies are relatively inexperienced in developing non-financial software. (Most loan portfolio software is developed by companies that also have developed complete accounting systems.)

Finally, **Outcomes modules,** designed to track data for measuring changes, such as job creation, income increases and client attitudes, are the least developed of the commercial offerings. In large part, this is due to the lack of definition of how to measure outcomes and what data, therefore, needs to be collected by the MIS. Until this issue is better resolved, it will be impossible for commercial software companies to develop comprehensive outcome modules.

[24] Whereas this may not necessarily be true for microenterprise assistance, loan portfolio software is developed for a much broader market than just microenterprise assistance, and in this broader market, financial services are more common than non-financial services.

6.3. Screen out clearly inappropriate products

At this point, you will need to start dropping software packages from further consideration in order to narrow down the list. The intent is to reduce the list from the 20 to 30 alternatives you have identified, down to 10 to 15 options that appear to merit further study.

This may not be easy, as you may find you have too little information upon which to base a decision. Remember, at this point you have not done any significant amount of systematic and substantive research. You have just been sifting through whatever information you have come across in your research.

To start this screening, you will need to develop a list of key criteria. These should not be related to anything but the most obvious technical issues, as you may not have enough information to make an accurate decision at this time. Your list of criteria, largely based on your work from Phase I, will likely include issues such as:

- "Must have an existing client base. We will not consider software (or 'vaporware') that is not already tested and proven."
- "Vendors must be responsive to our initial inquiries for information. If they aren't responsive now, how can we expect them to be after they already have our money and we encounter problems?"
- "Our absolute top-end budget is $xxx. We cannot consider any software costing in excess of this figure."
- "Our needs are pretty basic. We don't need the advanced sophistication — in fact, we cannot handle the *complexity* — inherent in highly advanced packages, such as those used by commercial banks."
- "We expect to be highly reliant on technical support. Therefore, we want to make sure that such support is available readily, and will not consider software applications used only in the international microfinance community without U.S.-based technical people."

Once you have this list for your own institution, you can use it to immediately exclude any alternatives that clearly do not meet your requirements. Do not worry about ranking or prioritizing your alternatives at this point. It is too early in the process and you don't yet have enough information to make a sound decision. The purpose here is solely to eliminate clearly inappropriate choices.

Case Study: Listing and Screening Potential Options

The task force, with the assistance of their MIS consultant, completed their research and developed the following list of possible commercial products.

[**Note:** At this point the case study does *not* follow the experiences of Women's Initiative. Instead the case study has been modified to better demonstrate the following steps in the selection process. Also, the example products included in this analysis are all fictitious. Reviews of actual commercial products can be found in the ***2002 FIELD MIS Software Review***.]

Product	Decision		Rationale
SBMIS	**Consider**	Reject	Popular with other institutions
Jobs!	**Consider**	Reject	Integrated system; looks solid in independent reviews
LoanMaster	Consider	**Reject**	Designed for commercial banks; too complex and expensive
Promover	Consider	**Reject**	International software; not supported well in the U.S.
Tracker	**Consider**	Reject	Good reviews in literature; looks suitable for us
Impact now	Consider	**Reject**	Very old program; not well supported for the future; company shaky
Micro Track	**Consider**	Reject	Looks very promising; good comments from users

Chapter 7: Phase II, Step 2: Initial Software Assessment

After reducing your list of options down to 10 to 15 candidates in Step 1, you'll now find your real work begins! This chapter is by far the longest and most complex of the Manual. But, stick with it! It will be worth it!

The challenge in this step is to do sufficient research about the alternatives to make a sound decision, but not invest an inordinate amount of time doing so. You will need to create and review functional comparison tables in order to make your decision. Hopefully your research in Step 1 will have identified some good sources of information from others who have already undertaken this task. (Don't overlook the ***2002 FIELD MIS Software Review*** as a starting point for your work!)

Therefore, this could be the step in which you invest the greatest amount of time and effort. On the other hand, based on your situation, needs and resources at hand, it may prove to be a quite straightforward process. For example, if you need only a loan portfolio module, you may find that much of the research has already been done for you, and you can quickly narrow your selection to the three to five options to consider in Step 3.

Phase II:
Assessing the Alternatives

Step 1: Generate and screen list of alternatives
- Research options (20-30 packages)
- Screen out clearly inappropriate options

Step 2: Initial Software Assessment (10-15 options)
- Group products by module
- Complete functional comparison tables
- Assess vendor support
- Assess each individual module
- Rate the "fit" for each product
- Eliminate inappropriate options

Step 3: Detailed Software Assessment (3-5 options)
- Thorough testing of demo software
- Discussions/negotiations with vendors

Step 4: Final decision and system strategy
- *Integrated vs. Hybrid?
- *Use commercial off-the-shelf software, or
- *Modify commercial software, or
- *Develop custom software

7.1. Group products by module

As you progress on your research, review any documentation you receive and summarize information you have in a simplified "Overview Matrix," as shown on the next page. Note that the data in the table is more complete than is typical at this stage in the analysis. Generally, it will require completing the analysis steps described in this chapter, in order to reliably complete the information in the table.

1. First of all, identify which modules you will need in your MIS (refer to Section 2.3.1 for an explanation of modules). Use the empty lines to add any modules you need that are not already identified on the list.

2. Fill in the “Our Needs” column using information from Phase I: Needs Assessment (see Section 4.5).

3. Identify one software product per column.

4. After studying documentation and reading whatever reviews and user comments you have found, rate as best you can the abilities of each product by module using this scale: “1” for a Basic level product, “2” for an Intermediate level product and “3” for an Advanced product. If you are unsure at this point, you can mark the product as “1-2” or “2-3.” The rating is only in general terms, and can be based upon your initial perceptions. You can revise your ratings as you become more familiar with the software. If the company offers an optional add-on module you might want to mark this with “Opt.”

5. Rate each product on the three dimensions of complexity as best you can, using the same Basic/Intermediate/Advanced scale as before (see Section 2.3).

6. Note any current users of the software. Knowing about the institutions that use the software can help you focus in on the software that is most likely to be suitable for your institution.

The companion document to this manual, *2002 **FIELD MIS Software Review,*** provides a version of the following matrix summarizing the various modules offered by a variety of companies.

Overview Comparison of Software Products

Product	Our Needs	SBMIS™	Jobs!™	Tracker™	MicroTrack™
Modules	Ratings are: 1: Basic; 2: Intermediate; 3: Advanced; N/A: Not Applicable; Opt: Optional				
Loan Portfolio	2		3	2-3	2-3
Demographics	3	3	3	2	2
Contact Mgmt.	2		2		3
One-on-One TA	3	2			3
Training Courses	2	1			2
Outcomes	3	1			2
Accounting	2			Opt	Opt
Complexity	Ratings are: 1: Basic; 2: Intermediate; 3: Advanced				
Breadth of Services	2	2	1-2	2	3
Depth of Information	3	1-2	2	2	2-3
Scale of Institution	3	1-2	3	2	3
Customers					
Software Currently Used by:		CDFU ACDU	CommVest	MnWorks	ARCLU DCJobs
Decision		To be completed at end of this step			

7.2. Complete functional comparison tables

In this initial assessment, the task force should carefully review all documentation provided by the software firm and all available detailed reviews of the software. It should focus on major issues of compatibility (such as types of financial products and interest-calculation methods supported), rather than on more technical details (such as procedures for calculating penalties), which are sometimes difficult to determine from basic documentation.

Still, areas of potential incompatibility should be carefully noted for later discussion with the provider during Step 3. Incompatibilities can sometimes be solved through undocumented features of the software or through relatively minor software changes. But, sometimes seemingly minor incompatibilities require a complete rewrite, meaning the system cannot be considered as a possibility. It is often difficult to know in advance which incompatibilities can eliminate a system from consideration and which can be easily addressed.

The following sections describe how to perform initial assessments by functional area and by module:

- Overall Features
    - Ease of Use
    - Reporting
    - Security
    - Integration
    - Technical Specifications
- Vendor Support and Technical Assistance
- Demographic Modules
- Loan-Portfolio Modules
    - Lending Methodologies
    - Loan-Pricing Options
    - Funds Management
    - Repayment-Schedule Options
    - Late-Payment Options
- Contact-Management Modules
- One-on-one Technical-Assistance Modules
- Training Modules
- Outcomes Modules

Each section will present some theory about the concept, often presenting different table structures and the relative strengths and weaknesses of different approaches. Each section also contains a functional matrix that allows each software module to be evaluated feature by feature (see example). The features listed in each table are explained in the body of the Manual.

Ease of Use	Explanation	Our Needs (Ess/Use/NA)	Product 1	Product 2	Product 3
Keyboard Data-Entry	System is designed to allow rapid keyboard data entry without use of mouse				
Data Validation	Most fields contain data validation to insure accuracy of data input				

To use the functional matrices, first fill in the "Our needs" column using the classifications of "Essential," "Useful" and "Not Applicable." This determination should be based on the findings during **Phase I: Defining Needs.**

You can then begin to fill in the columns for each product under consideration. It will not be necessary to complete all rows for all the products. You may find some products to be disqualified due to lack of an essential function early in the process. You can then eliminate them without needing to complete the analysis, unless you are considering extensive custom modification of the software.

7.3. Assess overall features

This first review section focuses on overall features of the software, i.e., aspects that are not specific to any one module of the system. These features are divided into the following categories:

- Ease of use
- Reporting
- Security
- Integration
- Technical Specifications

Each section provides a description of the category and a table to be used for documenting the features of that product.

7.3.1. Ease of use

Ease of use issues are generally not essential elements of the software, but they do generally improve staff productivity, shorten the learning curve for new users and significantly effect the overall experience of the user — either positively or negatively. The various aspects include:

Keyboard data-entry: Most Windows-based systems require extensive use of the mouse to move from field to field and enter data. While these input screens are visually appealing, they can slow down data-entry. For small volume data-entry, this may not be an issue, but if data-entry staff is inputting information off paper forms for long periods of time, a system that allows input primarily through the keyboard can significantly increase productivity. There is no reason why software cannot offer both. Frequently, occasional users will use the mouse more often (in lieu of remembering a sequence of keystrokes), while regular users will easily remember the keystroke combinations and appreciate the added speed in data-entry. Even better are systems that allow most data-entry to be keyed from the numerical keypad to the right of the keyboard.

Data validation: A chronic problem with information management is insuring the entry of valid and complete data. Carefully designed software can assist with a significant portion of data validation. Software can verify whether required fields have been left empty and prompt the user for an input. It can compare the entry to a range of values (such as valid date fields, realistic birthdates, appropriate minimum and maximum values, etc. It can also compare the value of one field against the value of another field, called cross-field validation. Depending on the field, the user can either be warned to confirm the entry, or can be prohibited from entering that particular value. In some cases, a supervisory override may be programmed in, e.g., if the loan amount exceeds the user's authorization level.

On-line help: Beginning users can benefit from an on-line help facility, ideally context sensitive, so that it opens up to the relevant section.

Printed users manual: When first learning a system, most users benefit from having a printed manual available. The best software documentation approach is to have two distinct manuals. First-time users benefit from a "getting started" document that explains the software in a logical sequence and provides examples and a tutorial. Intermediate users benefit from a second document, organized more along the lines of a reference manual, where they can turn quickly to the section relevant to the specific task they want to accomplish. Look for both.

User tutorial: Tutorials are an excellent way to get introduced to a new software package. They are most useful when no experienced users are around to give an overview, such as when you are evaluating a new software package for possible purchase. Tutorials should allow you access to the actual system and allow you to enter data under guidance. Tutorials which are essentially movies that play on your computer, showing someone else entering data, are less valuable.

Training courses: Even if staff is reasonably proficient in using software, training is generally advisable. Software vendors generally offer some sort of training on their software; however, each vendor will have his or her own policies about the cost of the training and what amount, if any, is incorporated into the purchase price of the software. Vendors may provide training by three different means:

- On-site training. The trainer comes to your office to train one or more staff. This is ideal, but generally quite costly.
- Internet-based training. This is remarkably useful and can be delivered at quite low cost, as all travel costs are eliminated. By use of screen-monitoring programs, the off-site trainer can watch the user work with the software and provide comments and corrections, either via phone or instant messaging. The trainer may even monitor more than one student at a time.
- Off-site training. Some companies offer training at their offices, requiring you and your staff to travel there.

Ease of Use	Explanation	Our Needs (Ess/Use/NA)	SBMIS™	Jobs!™	Tracker™	MicroTrack™
Keyboard Data-Entry	System is designed to allow rapid keyboard data entry without use of mouse	Use				
Data Validation	Most fields contain data validation to insure accuracy of data input	Ess	X	X		X
On-line Help	Provides pop-up help or Windows-style help file	Use				X
Printed User Manual	Software printed operations manual	Ess	X	X	X	X
User Tutorial	Documentation includes step-by-step user tutorial	Use	X		X	X
Training Courses	Vendor provides training	Use	X	X	X	X

7.3.2. Reporting

One of the most common weaknesses with MIS is poorly designed, inflexible reports. Very frequently, extremely useful data is collected and stored inside the system, but this information is not extracted in a useful form for staff to use in decisionmaking. Or, if the report does exist, staff has not been adequately trained to interpret and use the information.

Number of loan reports: It is tempting to compare software packages purely on the number of reports they generate, but this can be very deceiving. In practice, an institution will use only a handful of reports on a regular basis. It is better to have a few, well-designed reports than a large number of badly-designed reports. In addition, the issue of user-customization and filters can mean that what one software vendor lists as 10 reports can be listed as a single report by a different vendor. Thus, is it important to actually look over the types of reports and examples rather than simply to rely on reported numbers.

Number of other reports: The same comments apply as above.

Number of filter criteria: Good software programs will allow the user to select from a wide variety of filter criteria when generating a report. Examples include filtering for a specific range of dates, loan officer, branch office, loan fund or delinquency status. The more criteria available, the more flexible the system is without requiring the user to resort to the actual design of custom reports.

Customizable reports: Vendors understand that users have very different reporting needs, and that they cannot begin to anticipate every type of report a user will desire. Thus, many packages provide add-on software that can be used to customize existing reports or design new reports.

Can preview reports: It is very useful to have the ability to preview reports on the screen before choosing to print them. Most Windows-based programs provide this ability, and many of the DOS-based programs do, as well.

Reporting	Explanation	Our Needs (Ess/Use/NA)	SBMIS™	Jobs!™	Tracker™	MicroTrack™
No. of Loan Reports	Number of basic loan-report formats (not counting filtering criteria as unique reports)		1	4	7	17
No. of Other Reports	Number of pre-defined non-loan report formats		7	11	3	15
No. of Filter Criteria	Number of filtering criteria provided for reports		1	3	10	2
Customizable Reports	Includes a custom report writer	Ess	X	X	X	X
Can Preview Reports	Allows a screen preview of reports	Use	X	X	X	X

7.3.3. Security

Security issues will be of varying importance for different institutions, primarily due to the scale of the institution. Please refer to Section 2.3.3 for an overview of security issues.

Password logon: Opening the system requires some sort of password, even if it is a generic password entered by any user. This is the minimal amount of security to prevent entry by unauthorized users.

User logon: Systems with user logon require each person to be authorized by the system administrator and assigned a password. This provides much better control over access to the system, although still far from fool-proof. System logs can optionally be maintained, recording when a particular user logs on and off the system.

Force password change: Because passwords can often become commonplace knowledge around the workplace, some systems will require users to change their passwords on a regular basis.

Audit trail: Systems with user logon can use the information to maintain an audit trail. An audit trail records all new entries and changes made to the data, recording the identification of the user making those additions and changes. This is a high level of security, but can be extremely useful in tracking down data problems.

Data files protected: Opening an application generally requires a password logon. However, in many database systems, particularly older formats such as DBF files, the actual data files themselves are not protected in any way. They can be opened directly by a variety of different programs, viewed and even altered, leaving no audit trail. For some institutions, lack of security of the data files will require you to exclude the system from consideration.

Number of user security levels: Systems can build in different levels of user access for different staff positions. Levels can be established to view only data, enter certain types of data, approve loans, override interest and penalty charges, etc. There will also be a system administration level which allows the administrator to add new users, change customized drop-down lists, etc. Systems will generally have a minimum of three levels and up to about nine levels. Larger institutions will generally need more levels.

Security	Explanation	Our Needs (Ess/Use/NA)	SBMIS™	Jobs!™	Tracker™	MicroTrack™
Password Logon	Requires entry of a password to open the system	Ess		X		X
User Logon	Requires each user to have a user ID	Ess		X		X
Force Password Change	Require users to change passwords at a specified frequency	Use				
Audit Trail	Tracks changes made to records	Use		X		
Data Files Protected	Data files cannot be opened from outside of the system	Use		X		
Number User Security Levels	Number of different security levels for users	4		12		9
User-customized Security	Each user's security access can be customized	Use		X		X
Remote User Authentication	For dial-in users, or for Web-based software, thorough authentication is important	Use				

User-customized security: Some systems will allow the administrator to customize what areas of the system can be accessed by each of the different security levels. In some instances this customization can be done user-by-user, but generally the customization is for a specific access level.

Remote user authentication: If you expect to allow staff to access data via remote dial-up connection, or if you are planning a Web-based application, thorough security will be very important.

7.3.4. Integration

Integration issues for different modules were discussed previously in Section 2.3.1. Vendors are expected to provide smooth integration of modules which they provide. However, integration of modules from different vendors is generally lacking and will remain so unless vendors begin collaborating in an effort to build strategic alliances among their products.

In the absence of seamless integration, users are forced to create their own methods of integrating data. This generally takes the form of manual systems and ad-hoc Excel™ spreadsheets. Most MIS modules do provide at least some sort of basic exportation of tables into common formats, such as DBF or XLS files. However, there are generally significant limitations to the usability of such exports, particularly in systems with a large number of highly related tables. Exporting individual tables from the system and then importing those tables into Excel™ will often not give the degree of functionality that the user would hope for.

Perhaps the area of greatest interest and concern is the issue of integrating data from the loan portfolio module with the institution's accounting system, as the majority of institutions do not have modules provided by the same vendor. See Section 9.2.2 for further information on integration issues.

Link to accounting: Portfolio data can be linked automatically to either an accounting module provided by the same vendor or to another company's accounting package. Simply indicating that the software provides linkages does not mean that the linkages will be easy or reliable. You will need to review the procedure carefully to insure that it meets your expectations.

Export tables: Most applications provide a means to export tables into generic formats. You should review the formats to make sure they are acceptable. You should also investigate whether you can use queries to create customized temporary tables which you can then export. If so, you may be able to put together more useful information outside of the application than you would otherwise.

Integration	Explanation	Our Needs (Ess/Use/NA)	SBMIS™	Jobs!™	Tracker™	MicroTrack™
Link to Accounting	Portfolio data can be linked to other accounting modules	Use		X		X
Export Tables	Allows tables to be exported in a common exchange format	Ess	X	X	X	X

7.3.5. Technical specifications

It is important to research the technical specifications of the various applications, although this is an area in which managers will likely need the most guidance from technical experts. There is little advice we can offer in a manual of this level that would not either be too superficial, too technically dense or soon out of date.

For most managers, it would be best to simply collect the information about technical specifications and then, once the choices are narrowed down, review the various aspects with a MIS consultant. In order to get answers to these questions, it is generally best to simply request the answers directly from the vendor.

Please refer to Section 10.2 and Annex 5 for a basic overview of hardware issues.

Web-based application: Indicates whether the software is provided by an applications service provider (ASP), in which case staff uses the Internet to enter and retrieve data. Web-based alternatives were explained in Section 2.2.3.

Workstation RAM: Workstations are the computers that your staff members use. These are their desktop computers and laptops that are then connected into the network. Workstation RAM indicates the minimum advisable RAM requirement to run the system. Note that this figure is difficult to indicate reliably and does not consider "real life" situations where staff members also will have a number of other software applications open simultaneously. Often, doubling the minimum recommended RAM will serve as a rough estimate of an adequate amount of RAM.

Software language: Applications are developed using a software language, which is then compiled and distributed. Many times the development language will not be apparent to the user. The information is relevant here to help the MIS consultant determine the technical quality of the language (each language has advantages and disadvantages) and to determine if the language is still supported by a healthy company.

Database: This indicates the storage format for the data files. Different storage formats influence the access speed, the overall capacity of the system to hold large amounts of information and security issues. (Security aspects are explained further in Section 2.3.3).

Workstation Operating System: Indicates which operating systems staff members can have on desktop computers. Some applications may require users to have Windows NT/2000/XP.

Server Operating System: Indicates the operating system for the server, which can differ from that required for desktop computers. The server is the computer that runs the network. It contains the hard drive(s) where data is generally stored, allowing users at different workstations to share that data.

Networks supported: Indicates which network systems can be used in conjunction with the application.

Technical Specs	Explanation	Our Needs (Ess/Use/NA)	SBMIS™	Jobs!™	Tracker™	MicroTrack™
Web-based Application	Data is stored on the vendor's servers	NA				
Workstation RAM	Minimum RAM requirement for workstations	Max 64M	24M	128M	32M	64M
Software Language	Language used for development of the software application	NA	MS Access™	MS Access™	FoxPro™	Delphi™
Database	Format used for storage of the data files	NA	MS Access™	MS Access™	FoxPro™	Dbase™
Workstation Operating System	Required for desktop computers of users, e.g. Win 95, NT, etc					
Server Operating System	Required for the server; can differ from the OS on the desktop computers					
Networks Supported	Which networks are compatible with the application, e.g., Windows NT, Novell, etc					

7.4. Assess vendor support and technical assistance

Deciding on a MIS is not like deciding on an Internet browser or word processor! With a MIS, you're not buying software; you're entering into a long-term — and often close — relationship with a company. You should consider it equally as important to research the company's support capacity, history, financial viability, corporate culture and commitment to the product you are evaluating, as it is to review the actual features of the software. Few users ever have any direct contact with Microsoft, but you will soon be on a first-name basis with most of the staff of the company that provides your MIS software.

The items in this table are a mix of objective and subjective indicators. Objective indicators include issues such as the number of employees and number of clients, and tend to indicate the health of the company, but may create a bias toward larger companies. Subjective indicators are included as a means to judge the culture of the company. Most microenterprise advocates will agree: Bigger is not always better. And, unfortunately, even well-designed software can be frustrating and even worthless, if not backed by a reputable company that places a high value on ongoing customer satisfaction, and isn't simply interested in closing sales.

Because of the subjectivity of some of these indicators, you will generally not find this information contained in official independent reviews. This is information you will likely need to compile yourself, based on your dealings with the company and any opinions you can pick up in discussions with other clients using the software. Despite the challenges in obtaining this information, do not avoid doing so, as it will be critical for your final decision.

Case Study: Picking the Wrong Vendor!

Even when institutions invest a considerable amount of time, effort and money, they can end up with nothing to show for it. In our research we found a number of institutions that expressed grave disappointment and frustration because they felt the vendor did not fulfill the requirements of the contract. Some of these experiences had led to lawsuits. In some cases, the software was functioning, but not up to expectations. For others, the software was completely non-functional. Horror stories abound, but in many cases the disaster could have been averted by picking the right vendor from the beginning.

However, to be fair, failures are not always fully the responsibility of the vendor. In many cases, both parties are to blame, and the problems are due to poor communication and poor definition of information needs, the key areas that this Manual is attempting to address (see Section 1.3).

Number of employees: This number can be deceptive in that some companies sell multiple products and some are dedicated to only the software you are considering. Still, this can provide a general indication of the size and stability of the company.

Number of clients: Again, some software companies may provide a range of products. Since your interests are in certain specific modules, you should ask for the number of clients currently using the software. Some vendors may report this as installations, as one contract with a major client may result in a large number of installations in branch offices or affiliates.

Client references: At this stage, it is not necessary to check with client references. However, it would be useful to ask the vendor about their willingness to provide a list of references. Their freedom to do so and the extent of the list may be indicative of the company's transparency and its rapport with its existing client base.

Demonstration on CD-ROM: Does the company provide a functional demonstration version of its software on CD-ROM? Vendors will often limit the length of time the demonstration can be used or the number of clients that can be entered into the system. But, the demonstration should be fully functional in other respects to give you a reasonable means of judging the software.

Demonstration download from Web: Some companies may provide for downloading their demonstrations from the Web, sometimes from a private Web page that they will provide to you after you contact them to express your interest. In some cases, the software is so large that the company prefers to send you a CD-ROM.

Demonstration by remote access: Because of complexities of installation processes and configuration setups, some vendors may prefer to make an appointment and provide a demonstration of their software via remote access. Such sessions can be remarkably effective. Via the Internet, the vendor can show you exactly how the software appears on the screen, while simultaneously talking with you via phone. In most cases, control of the program can be passed to you, allowing you to navigate through the menus and enter data under guidance by the vendor. There is, however, a limitation to remote access demonstrations. They generally serve as a good introduction, but are not long enough to allow you to really evaluate everything about the program. And, you generally won't feel in a position to enter sample data from your own institution or to really explore the system, because the vendor will generally be leading the process. You will do better if you can ask for some extended, independent time with the system without the vendor monitoring. Another limitation is that you will not be able to print reports, though you should be able to view them on-screen. In summary, remote access demonstrations are a good starting point, but if you are serious about the system, you should inquire about getting a functional demonstration installed at your business.

Demonstration slide-show: Some vendors provide "slide-show" demonstrations that are essentially screen clips of how the program works. These are helpful for initial overviews, but do not take the place of a functional demonstration. You will still need to obtain a real demonstration of the program before drawing any conclusions about the software.

Corporate "transparency": As explained earlier, this is a subjective issue, and one which generally you will not find documented in independent reviews.

Company Information	Explanation	Our Needs (Ess/Use/NA)	SBMIS™	Jobs!™	Tracker™	MicroTrack™
Year Company Established			1997	1957	1998	1993
Number of Employees			2	70	4	15
Number of Clients	Number of clients currently using the module(s) reviewed in this report		20	40	160	160
Demo on CD-ROM	Vendor provides fully functional demo on CD-ROM		X	X	X	X
Demo Download from Web	Vendor allows fully-functional demo to be downloaded					
Demo by Remote Access	Vendor provides demonstrations of software by use of remote access software			X		X
Demo Slide-show	Vendor provides a slide-show that gives an overview of the software					
Corporate "transparency"	Vendor gives clear answers to all your inquiries					
Client References	Vendor provides an exhaustive list of clients using his/her products					

7.5. Assess demographic modules

In our non-profit work, we generally have called this area "demographic information." In business, and in MIS design, this data is typically referred to as "customer information." The demographic module — often referred to as the Customer Information File, or CIF, by programmers — serves to track who the client is, as well as other information of interest about the client. Typical information might include customer name, customer contact information, and socioeconomic and demographic information of interest to the institution for determining customer eligibility. Note that much of this information is also of interest for tracking outcomes, and therefore, there can be overlap between information contained in this module and information contained in or required for the outcomes module (see Section 7.8 for further information about outcomes modules).

Virtually every software product incorporates some semblance of a demographics module, though it is usually seamlessly integrated into their other modules and not actually presented as a distinct module.

7.5.1. Principal structural issues

Whether you are evaluating your existing data or designing a new client-tracking system, the principle areas of concern regarding the demographics module have to do with what data is stored and how it can be retrieved. Specifically, include the following:

1. Who (or what) is being tracked?
2. At what point is client information entered into the system?
3. How are client and business information related?
4. Who receives services — clients or businesses?
5. How much flexibility is there in the definition of what characteristics can be tracked?
6. What demographic information can be tracked over time in order to analyze outcomes?

These points are covered one-by-one in the following sections. The information in this section is presented in a fairly comprehensive fashion, as this is one of the more difficult areas for which to find satisfactory software.

1. Who (or what) is being tracked?

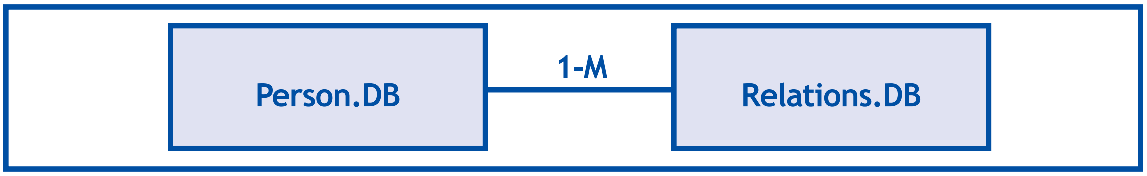
Does the system consider the client to be a business or a person? For example, some systems oriented toward providing services to larger businesses consider the business to be the "client" and do limited tracking of the actual individual(s) behind the business, which most microenterprise agencies will find unacceptable. The approach you select, treating the client as a business or as a person, therefore, will influence the types of demographic fields in the system.

Some systems also allow the CIF to store information not only on clients, but also on broader classifications of individuals, such as volunteers, client references, donors, staff persons and guarantors. Such an approach allows names entered into the system to be identified with one or more of these classifications, or "relationships," to the institution.

More advanced design approaches have incorporated sophisticated means to create a variety of types of relationships between two or more individual records in the CIF, where each record represents an individual or an entity such as an agency. Such an approach allows identification of relationships such as:

- "John Smith is guarantor of Adam Jones."
- "Sue Brown is loan officer for Adam Jones."
- "Adam Jones is a business partner with Cathy Larson."

This approach makes the CIF much more versatile, but requires more complex programming. The key is to create a Relations table that will store all the various relationships between people and between people and entities such as agencies.



The schematic below shows the table relationships for one means for implementing this approach. (The "1-M" signifies a "one to many" relationship between data in the two tables, i.e., one person may have many roles or responsibilities the system can thereby capture.) Note that start- and end-dates are not included in order to keep the example more straightforward. You will note there are actually two tables required:

Table 1: **Relations Table:** This table contains relationship information for each two-way relationship you want to track — one row per relationship:

ClientID The unique five-digit number your agency assigned each individual or entity.

Related to Again the unique five-digit number your agency assigned each individual or entity.

Relation The relationship between the ClientID and the Related To fields. As an example, record one indicates that ClientID "10500" is related to "99001" and the relationship is that "10500" is a client of "99001."

Person.DB

Record	ClientID	Name
1	10311	John Smith
2	10500	Adam Jones
3	10501	Cathy Larson
4	30220	Sue Brown
5	99001	MicroAssist

Relations.DB

Record	ClientID	Related To	Relation
1	10500	99001	Client of Institution
2	30220	99001	Employee of Institution
3	30220	10500	Advisor
4	10500	30220	Advisee
5	10311	10500	Guarantor
6	10500	10311	Guarantee
7	10501	99001	Client of Institution
8	10501	10500	Business Partner
9	10500	10501	Business Partner

Table 2: **Person Table:** This table contains a row with more detailed information for each individual or entity you want to track:

ClientID The unique five-digit number your agency assigned each individual or entity.

Name The full name of the individual or entity.

By comparing the information in the two tables, you can find all of the example relationships given in the previous paragraph. An advantage of the structure of Relations.DB is that for any given person, the system can readily show all existing relationships for that person by using a filter on the ClientID field.

2. At what point is client demographic information entered into the system?

There are two fundamental approaches:

- Input demographic information only when a person is officially considered a client, e.g., when he or she receives a loan or when they register for a course.
- Input demographic information for prospective clients, e.g., phone or drop-in inquiries about the services. Registering prospective clients can assist staff in follow-up activities, help generate useful market analysis and provide a more custom focus as individuals are not asked to provide the same information on multiple occasions.

The second option provides more versatility, but requires a means to identify when someone actually is considered a client. This requires a field in the client table and a means to manipulate that field. The same field may also be used to differentiate "active" clients from "inactive" clients who no longer receive services.

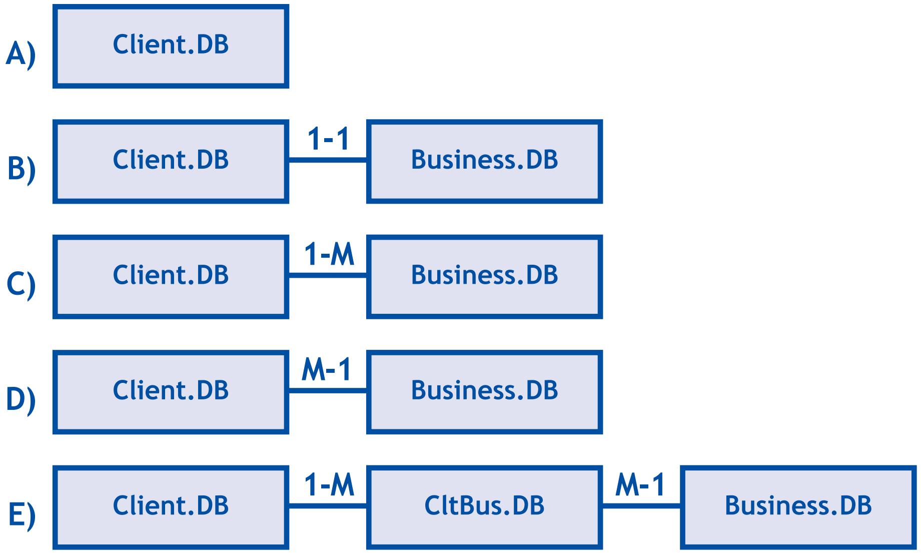
3. How are client and business information related?

These are profoundly important design considerations. Once a system has been designed and developed, it is often a very complex operation to modify the approach. These issues are addressed in the table layout in the system. The fundamental issues are:

- Is there only a single table to hold client and business information? If so, this is very limited in flexibility. There are fields for client name, address, birth date, etc., and there are fields for business name, business address and age of business. The assumption is one client (owner) has one business, and one business has one owner.
- Are there separate tables for client and business? If so, can a client have multiple businesses? Can a business have multiple owners (i.e., clients)?
- If there are separate tables, are services (training, loans, etc.) linked to the client table or the business table?

The figure below portrays an array of five options for managing and relating client and business information. The right-hand column provides brief comments on the usage and limitations of each option, which is fully described below.

Possible Relationships for Client and Business Information



Comments

All information about the client and business is stored in a single table, using a long string of fields.

Information is split between two tables, but functionality is identical to (A) because of the 1-1 relationship.

A client can have one or multiple business activities, but each business only one owner.

A business can have one or multiple owners, but each client only one business.

Any client can have multiple businesses, individual or jointly owned. Any business can have a single owner or multiple owners.

Scenario A is very basic and actually fairly common in commercial systems. All information about the business and the client is stored in a single table. The client can have only one business, and that business cannot change over time as the client continues to receive services from the institution, nor, can the business have more than one owner.

Scenario B is functionally identical to Scenario A, but the fields have been split up between two tables. While the presence of two tables implies more flexibility, the one-to-one (1-1) relationship eliminates this potential.

Scenario C separates the information into two tables and allows a single client to have multiple businesses due to the one-to-many (1-M) relationship. Each record in Business.DB is linked via a client ID to the Client.DB table. A client can either have multiple businesses simultaneously, or evolve from one business to another over time without losing (overwriting) information about the previous business. The approach is suitable for most microenterprise programs.

Scenario D is the reverse of Scenario C in which a business can be identified as having multiple owners, as would happen with agencies supporting small business, but interested in tracking demographic data about the owners.

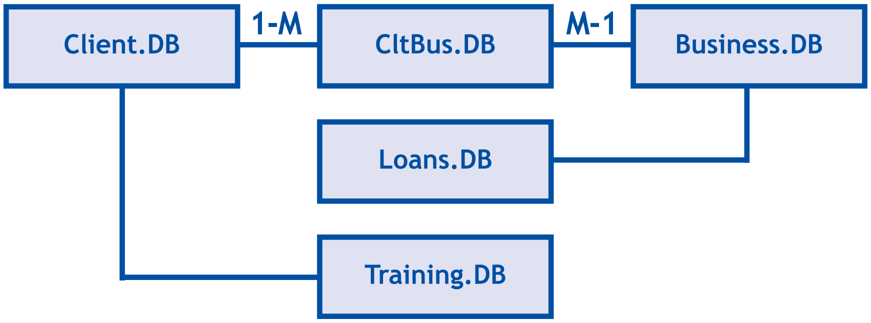
Scenario E is clearly the most flexible approach, suitable for nearly any situation. The approach uses a "relation" table called CltBus.DB to allow a many-to-many (M-M) relationship between clients and businesses. Because of the many-to-many capability, a client can have multiple businesses, and any business can be associated with multiple clients. The tradeoff is the greater complexity inherent in designing this system and the entry screens that allow the user to establish and maintain these relationships.

4. Who receives services — clients or businesses?

Do services go to individuals or the businesses they own? What happens if an individual has two businesses that receive assistance? What happens if a business your organization assists has two or more owners?

The table structure defines this issue by linking any activity tables, e.g., Loans.DB or Training.DB, to either the client table or the business table. In the example below, clients can have multiple businesses and businesses can have multiple owners. However, loans are linked to a single business, whereas training activities are linked to a single individual.

In this particular area, a significant mismatch between the operational procedures in your institution and the design approach of the system may mean you need to eliminate this system from further consideration, as modification of this area is often too complex to consider as a "customized software" option (see Section 9.4 for more information about modifying commercial software).



Any client can have multiple businesses, individual or jointly owned. Any business can have a single owner or multiple owners.

Loans are identified to a single business.

Training activities are linked to a specific individual.

5. How much flexibility is there in the definition of what characteristics can be tracked?

This relates to two aspects described in Section 2.3.2 — the number of user-defined fields and the ability to customize drop-down lists. Ideally, each table in the demographics module (e.g., Clients.DB and Business.DB) would have a healthy number of user-definable fields, some of which can be configured as text, number or date, and some of which can be linked to customizable drop-down lists. In addition, the drop-down lists themselves should be customizable by the system administrator. (Allowing the user to modify the drop-down lists can often create problems of incorrect or inconsistent entries. It is best that this activity be limited only to the system administrator.)

6. What demographic information can be tracked over time in order to analyze outcomes?

As emphasized in Section 2.3.2, table structures need to be carefully designed to avoid writing over important information when that information changes in the future. A detailed explanation of how different structures can influence storage of demographic data can be found in Section 7.8.

7.5.2. Functional matrix

The following issues are important when evaluating demographic modules:

Who is the client: Indicates whether the software considers the individual(s) who owns the business or the business itself to be the client. Software oriented toward funding medium to large businesses will tend to focus on the business as the "client."

Can include non-clients: Is the software structured so that data may be entered on individuals or companies that are not expected to be clients, such as referrals, volunteers, donors, etc.? The advantage of such a system is having all contacts in a single location. Such software typically then uses "relations" to indicate which of the records in the table belong to clients (see below).

Allow identification of relationships: Versatile systems will allow the establishment of "relationships" between different entities in the Clients.DB table. The table will include clients and non-clients, co-signers, employees of the institution, etc. Such a system is very powerful, but more complex to develop and maintain.

Can include pre-clients: Some systems allow input only once the person is considered a client. Other systems allow the tracking of potential clients, in which case there needs to be a way to indicate at what point the person becomes a client. The advantage of this approach is to enable the system to act more like a "contact manager" with which staff can track leads. The system also contains potentially valuable information about how many inquiries translate into clients.

Multiple clients per business: The system allows each business to have one or more owners, as described earlier.

Multiple businesses per client: The system allows each client to have one or more businesses, as described earlier.

Linking of activities: Does the system link activities, such as loans and training, to the clients or to the businesses? These issues were described in Point 4 on the previous page.

Number of pre-defined contact fields: The number of fields dedicated to contact information (address, phone, etc.) per client, not including user-defined fields. A higher number will indicate a greater degree of detail and imply how sophisticated the demographics are.

Number of pre-defined demographic fields: Number of fields dedicated to demographic information on the client (not the business). This includes fields such as gender, birth date, income level, etc.

Number of user-defined fields: Number of user-defined fields dedicated to client or business information. Larger numbers will imply a more versatile system in which you will be able to supplement the pre-defined fields with additional fields of interest to you. The actual usefulness of the fields will depend on the field type (numeric, text, date or drop-down list).

Number of user-customizable drop-downs: Number of drop-down lists for demographic data which may be user-customized. These will greatly enhance the usefulness of the user-defined fields mentioned above.

Demographic Module		Our Needs (Ess/Use/NA)	SBMIS™	Jobs!™	Tracker™	MicroTrack™
Who is the Client?	Is the focus point of the software the individual(s) who owns the business or the business itself?	Indiv	Indiv	Bus	Indiv	Indiv
Can Include Non-clients	Allows input of information for individuals/ companies not expected to become clients, e.g., referrals, volunteers, donors, etc	Ess		X		X
Allow Identification of Relationships	Individual entries in Clients.DB can be linked to other entries in a variety of different relationships	Use	X			
Can Include Pre-clients	Allows input of information on prospective clients	Ess		X		
Multiple Clients Per Business	Allows more than one business record to be linked to a client record	Use	X			
Mulitple Businesses Per Client	Allows more than one business record to be linked to a client record	Use	X			

Demographic Module		Our Needs (Ess/Use/NA)	SBMIS™	Jobs!™	Tracker™	MicroTrack™
Linking of Activities	Area activities (loans, training) linked to individuals or businesses?	Indiv	Indiv	Bus	Indiv	Indiv
No. of Pre-defined Contact Fields	Number of fields dedicated to contact information (address, phone, etc.) per client, not including user-defined fields		11	8	15	11
No. of Pre-defined Demographic Fields	Number of fields dedicated to demographic information on the client (not the business)		12	0	7	5
No. of User-defined Fields	Number of user-defined fields dedicated to client or business information		0	0	0	0
No. of User-Customizable Drop-downs	Number of drop-down lists for demographic data which may be user-customized		2	0	0	3

7.6. Assess loan portfolio modules

A loan portfolio system captures information and generates reports on the performance and status of all loan activity. It is an exceedingly complex module and must be carefully designed to fit the institution and the financial products it offers. The portfolio system needs to work potentially with an array of loan products, each of which may operate under substantially different rules, such as for interest rates, interest calculation methods, maximum allowable amounts and terms, definition of overdue payments and eligible collateral.

Because of the importance of loan portfolio modules, this section is relatively long. The first three sections which follow will discuss the issue of complexity in portfolio systems and the issue of linking loans to either the client or the business tables, and then present some typical table structures used in portfolio systems. The remaining sections will then analyze portfolio modules by functional area: lending methodologies, loan pricing options, funds management, repayment schedule options and late payment options.

7.6.1. Complexity of portfolio systems

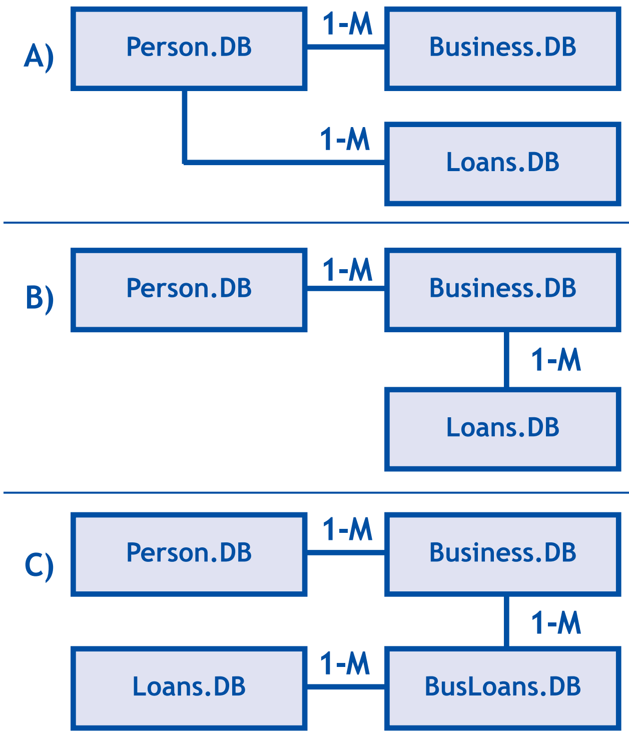
Most of the complexity in developing a loan portfolio system is due to two key issues: generating the contractual repayment schedule and determining what to do if the client does not follow that schedule. The surprising variation in the way different microfinance institutions — and even different loan products within a single microfinance institution — treat these matters is at the root of many incompatibilities between off-the-shelf portfolio systems and institutions' practices.

Much of the variation is due to the tendency that microfinance institutions have had to adopt approaches that are simple to implement and make sense at the time, rather than approaches that make sense from a finance viewpoint or resemble commercial banking standards.[25] Once adopted, practices are hard to switch because changes in loan treatment cannot always be implemented retroactively.

In response to this diversity of approaches, many companies have invested considerable effort in building in a wide variety of configurable options in these two areas in an effort to satisfy the diverse requirements of their clients. However, this approach makes the software both complex to develop and very difficult to maintain. Even small, seemingly minor, changes in one area of the programming can have unintended consequences in another area of the programming. Comprehensive testing and debugging with such complex coding can be a near-impossible task.

7.6.2. Linking loans to clients or businesses

As was explained in Section 7.5, there are several approaches for linking products and services to either the client or the business (assuming the system actually distinguishes between the two). The figure below shows three fundamental approaches for the linkage of loans to client-business data.



Comments

A single individual can have multiple loans. These loans cannot be shared among more than one individual, nor can they be linked to a particular business.

A single business can have multiple loans. These loans cannot be shared among more than one business. However, if the business can be linked to more than one owner, these owners can, in turn, be linked to the loan.

A single loan can support multiple businesses. In addition, if each business has more than one owner, all owners can be linked to the loan. Also supports the case where a single owner has multiple businesses supported by a single loan.

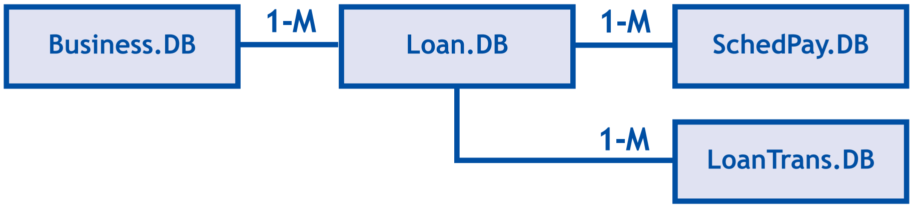
[25] Note that in international microfinance, this issue is far more complex than in U.S. microfinance, due to a multitude of operational variations that have evolved in various countries. One commercial package used in the international market offers more than 20 different methods for calculating interest, and still does not satisfy the needs of many microfinance institutions. And, not only do portfolio modules need to accommodate this extremely wide range of operational practices, but also they need to track many thousands of loans because of the very large scale that many of these institutions have reached.

- In option A, loans are linked to a single individual. If the individual has more than one business, the system does not identify which business was financed. In addition, a single loan cannot be linked to more than one person.
- In Option B, the linkage is to the business. A single business can receive multiple loans, but any one loan cannot be used to fund more than one business. If the system allows a single business to have multiple owners (a "one-to-many" relationship between Person.DB and Business.DB), then this approach does allow a loan to be traced back to more than one individual. Of the three options, this approach is likely to meet the needs of most microenterprise programs.
- Option C is the most complex and allows a single loan to support more than one business. These businesses can either have a single owner, different owners for each business or even multiple owners for each business. The Busloan.DB table allows a many-to-many (M-M) relationship between loans and businesses.

Note that of the three approaches, only Option A allows the provision of loans to individuals rather than businesses. Thus, institutions providing personal loans would need to select a system structured with a direct relationship between Person.DB and Loans.DB, whereas institutions focused on businesses and indirectly on clients would need to choose systems designed around options B or C. This is actually an interesting example of how difficult it is to develop a single system that is optimal in all situations.

7.6.3. Typical table structures

There are many legitimate approaches to table design for loan portfolio modules. Some can be extremely complex. The example shown below highlights a common approach, showing only the most essential tables. A complete system may contain a variety of other tables for look-up functions, to define distinct loan products, document collateral, identify loan purpose or to provide a number of other features.



Either the Person.DB or the Business.DB can be related to the Loans.DB table via a one-to-many relationship, meaning that a single business can receive more than one loan. Each record in Loan.DB, therefore, represents a single loan. In turn, each loan can then be associated with multiple records in an optional SchedPay.DB table, which generates a record for each scheduled payment, providing the date and the amount of the payment. Note that some systems will not generate records for this table, but instead rely on rules and fields in Loan.DB to generate the repayments "on-the-fly." In addition, there is generally a LoanTrans.DB table which contains one record for every transaction related to the loan, such as a loan disbursement or repayment. (In some cases, records may also be added to indicate a missed payment or to accrue interest to the account.)

Each of these three tables can have a large quantity of fields. Loan.DB will generally contain the specifics of the loan approved: date approved, amount approved, term, repayment frequency, maturity date, interest rate, loan funding source, collateral, loan purpose and penalty method. Sometimes the table will include some outstanding balance of principal and interest and some delinquency information, but in other systems this information will be stored in LoanTrans.DB.

The SchedPay.DB table, if used, usually has the least number of fields but the greatest number of records. Fields include loan reference number, payment date, scheduled payment amount and sometimes a "Payment Made" field. One record is created for each scheduled payment. In some systems, these records will be generated only for loans with irregular payment schedules; standard payment schedules are worked out using formulas and the basic information contained in Loans.DB.

The LoanTrans.DB table is the most important table from an accounting standpoint. It must provide a detailed record of every monetary transaction related to the specific loan, including disbursements, repayments and in some cases accruals. It must provide for an audit trail and enable information to be summarized for automated or manual transferal to the accounting system. Basic fields include loan reference number, transaction date and transaction amount, allocated to principal, interest, penalty and "other." In some systems, accrued amounts, outstanding balances, and overdue amounts and delinquency calculations are also stored in various fields in this table.

7.6.4. Evaluation by functional area

Because of the complexity and importance of loan portfolio management, an institution assessing software for this purpose needs to give careful consideration to the following areas:

- Lending Methodologies
- Loan Pricing Options
- Funds Management
- Repayment Schedule Options
- Late Payment Options

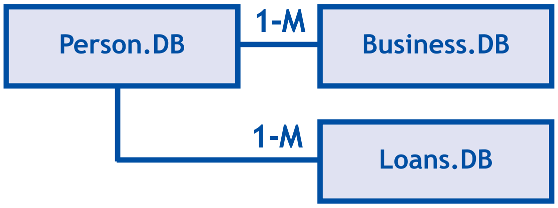
The remaining sections analyze each of these areas.[26] Numerous as they are, the issues included in the matrices do not exhaust the areas needing examination. Institutions will need to augment the points listed with any other specific procedures they use in their institutions.

Lending methodologies

The influence of lending methodology on software design is closely related to the issues of client and business information described in Section 7.5, **Assessing the Demographics Module.** Depending on the lending methodology, loan information needs to be related to one or more clients or one or more businesses. The relationship of the Client.DB and Business.DB tables will determine what is possible.

[26] The following sections incorporate some ideas previously expressed in the *CGAP MIS Handbook,* 81-86.

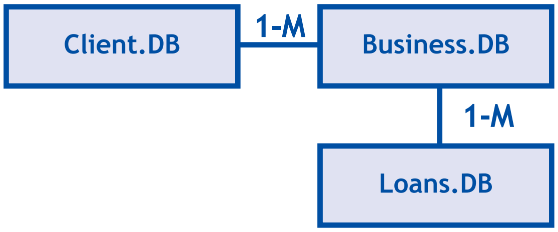
A)



Comments

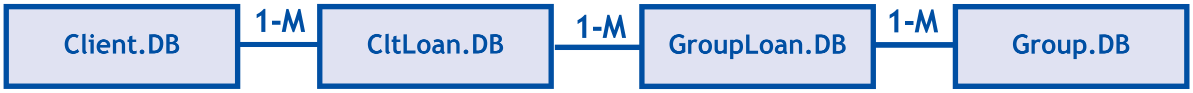
A single individual can have multiple loans. These loans cannot be shared among more than one individual, nor can they be linked to a particular business.

B)



A single business can have multiple loans. These loans cannot be shared among more than one business. However, if the business can be linked to more than one owner, these owners can, in turn, be linked to the loan.

C)



The above structure allows client loans to be tracked by individual and tied to specific group loans. In addition, clients can swich membership from one group to another over time.

Options include:

Individual lending: This is the most straightforward and common lending methodology, in which a single client is linked to a specific loan. The typical table structure is shown in Option A, though Option B would also be suitable.

Partnership lending: In this methodology, more than one person can be responsible for a single loan if they are co-owners of the business being financed. This relationship is shown in Option B.

Peer lending with individual loans: This methodology was developed in international microfinance and is associated with Grameen Bank in Bangladesh and ACCION's programs in Latin America. It became a popular alternative for U.S. microenterprise lending in the 1990's, but has diminished in use in recent years. In its most common implementation, each individual in a group gets a loan of his or her own (similar to "individual lending"), but the group members also have a level of responsibility for one another. Payments and balances are tracked for each group member's individual loan, but each member is also legally responsible for the delinquency of any other group member. In essence, these are individual loans with a concurrent responsibility as co-signer for other borrowers. Option C above shows a table structure necessary for allowing group composition to evolve over time as some groups disband and clients move to a different group, a reality that must be anticipated in the design of the system.

If lending volume is low, this methodology can be managed tolerably by a system that permits relationships among different records in the Clients.DB table, such as described in Section 7.5. In such an approach, each member of the group must be entered as having a co-signer relationship with all other group members. When doing a consultation on any particular individual, her loan balance can be displayed directly and the balance of all other group loans can be identified indirectly. However, in organizations with a large volume of group loans, this process will be tedious and inadequate, and it will be advisable to seek out a system that has been specifically designed to handle group lending. As in many other instances, adaptation of an existing system to handle peer lending can be extremely complex and costly.

Peer lending with a single group loan: In this methodology, similar to the "village banking" concept originally developed in the international microfinance community by FINCA , the institution makes a single loan to a group of people, who then split the loan amongst themselves and accept responsibility for collecting the individual repayments, which they submit to the institution as a single payment. In a MIS, this can be managed by the table relationship of Option B, identical to the "partnership lending" option described above. There is one significant limitation in that the record in Loan.DB monitors the amount loaned to the group but not the distinct amounts received by each individual. This would require either an additional table that would store the ClientID, LoanAmt, Date and LoanReference number, or use of the structure shown in Option C.

Participation lending: A number of U.S. development finance institutions are working with "participation loans," where the financing for a client/business is distributed among a number of development finance institutions. This is clearly an approach useful with larger loans (e.g., over $50,000) in order to distribute risk, and is not likely to be used in microfinance. Such loans could be tracked by each institution recording only the value of its contribution to the overall loan, but in practice, institutions prefer to record the entire loan amount received by the business in order to have an accurate financial picture of the business.

Lending Methodologies	Explanation	Our Needs (Ess/Use/NA)	SBMIS™	Jobs!™	Tracker™	MicroTrack™
Individual Lending (e.g, 1 or more loans to 1 client)	Most typical and straight-forward methodology.	Ess	X	X	X	X
Partnership Lending (e.g., 1 loan to 2 clients)	A loan record can be linked to multiple records in the client table.	Use	X			X
Peer Lending with Individual Loans (e.g., 4 loans to 4 clients)	Multiple records in the loan table can be linked to multiple records in the client table, even though there is a primary link of one client to one loan.	NA				

Lending Methodologies	Explanation	Our Needs (Ess/Use/NA)	SBMIS™	Jobs!™	Tracker™	MicroTrack™
Peer Lending with a Single Group Loan (e.g., 1 loan to 4 clients)	A single record in the loan table can be linked to multiple records in the client table. Client-level detail on the loan is not maintained.	NA				
Participation Loans	The business receives a loan funded from a group of lending institutions. The system tracks the overall loan and the funders' portions.	NA			X	X

Loan pricing options

We often have complex approaches to pricing our loans, combining mixtures of interest charges with fees, commissions, insurance payments, and even rebate and payment incentive systems. This makes designing the pricing calculations portion of the portfolio module quite complex.

Even basic interest calculations are not as straightforward as they may first seem. Many minor factors must be considered to insure that a portfolio system will operate as expected. Two main functions need to be verified. Firstly, how are interest payments calculated in a standard repayment schedule, that is, when the client repays exactly as requested? Does the system generate a repayment schedule that accurately reflects the way used by the institution? Secondly, how is interest calculated when repayment deviates from the schedule? If the client pays early, is she charged less interest? If she pays late, is she charged more? If the client's repayment date falls on a holiday, is she charged more interest if she pays on the following day? If clients are mailing in payments, meaning payments may come in a bit early or late, is there a flexible period whereby interest calculations do not vary?

In the needs identification phase, you should have carefully documented your loan pricing system. In reviewing your operational procedures you should have clear answers to what functionality you need to have with regards to the issues highlighted in the matrix on page 90. In addition, you may have identified other unique approaches used by your organization that you will need to incorporate into this matrix.

It is noteworthy that these issues do not have a dramatic influence on the table structure of the system. The various options may influence a few of the fields in some tables. Otherwise, the capabilities and features of the portfolio module are not easily identifiable from a review of the tables. Rather, the various calculation options are performed in the software code. Since source code is almost never shared with the client, the supported approaches and details of the calculations will need to be determined from the system documentation and interviews with the vendor. However, in some cases you will also be able to determine whether the portfolio module supports your needs by issuing a test loan and seeing how the software handles the calculations and what override options are presented on-screen.

The following comments and questions raise some of the key issues addressed in the matrix below:

Fixed or floating rate interest: Is the interest rate set for the term of the loan? Can it be adjusted at any point, or is it a floating rate pegged to an external index? How do rate changes affect the repayment schedule? Note that changing the interest rate midway through a loan can be problematic for most systems. As interest rates change, so will payment amounts.

Compounded interest: Is interest charged against loan principal only (the generally accepted approach) or is interest also charged on interest due, resulting in a higher, or compounded, interest fee.

Collection of interest: Is interest collected with each repayment (the generally accepted approach), or at loan disbursement, resulting in a higher financial cost to the client?

A 360/365-day payment year: Is interest calculated on a 360- or 365-day year (or on another system)? Although the effect is minor, the base number of days in the year affects the amount of interest charged.

Specify interest rate: Is the interest rate the same for all loans in a product category, or is the rate established for each individual loan at loan approval? Some software packages are designed to require that all loans for a product definition be charged the same interest rate. If interest rates can be set for each loan, what authorization is required to set the interest rate on a loan outside of a specific range? Is an audit trail kept of who has input the interest rate for the loan? (This can be an area for potential fraud.)

Interest calculations for delinquent loans: How is interest calculated if the client does not pay according to the repayment schedule? More accurate systems calculate interest on the number of days since the last payment, but many institutions simply charge interest for one payment period (one week or one month) even if the client is paying early or late, because the client knows how much is due, and it doesn't vary if the payment is early or late by a few days. Note that this is a common practice with loans in the U.S. that have payments mailed in. If the payment arrives a few days early or late, the interest charged is as if the payment arrived on the precise date the payment was due.

Accruing interest: Does the system allow interest to be accrued on loans? If so, when is it accrued — daily or at month-end? Is interest accrued on delinquent loans? At what point is accrual suspended or reversed? Systems which accrue interest can overstate the institution's income if the loan portfolio quality is poor, because income is reported on the financial statements that will never actually be collected from the client. In these cases the interest receivable will need to be aged and adjusted periodically in the same way in which the principal due is aged.

Up-front commission: Are up-front fees or commissions charged on new loans? Are they set amounts or percentages? On what base are they calculated? Does the percentage vary for loans of different size or is it constant? Is there a minimum or maximum limit on the size of the fee? If there are multiple disbursements for a loan, are the up-front fees assessed on the entire amount or on each disbursement? For up-front fees, it is normally best to have the system allow manual calculation and entry of the fees. That allows the greatest flexibility for changes in approach later.

On-going commission: Are on-going fees or commissions charged on loans? How are they determined? Are they accrued? On-going fees generally should be automated, since they will affect so many transactions. Note that many portfolio modules do not support automation of on-going fees.

Overriding interest charges: Many institutions give management the discretion to forgive interest charges or reduce the amount charged, on a case-by-case basis. Does the software support this? What safeguards are in place regarding adequate authorization level and audit trails?

Insurance charges: It is getting common to have loans insured against ill health or death. The insurance premium is generally paid by the client as part of his or her monthly loan payment. For institutions implementing loan insurance, having the process integrated into the software will save a great deal of work.

Loan Pricing Options		Our Needs (Ess/Use/NA)	SBMIS™	Jobs!™	Tracker™	MicroTrack™
Fixed Rate Interest	Interest rate does not vary during the term of the loan	Ess		X	X	X
Floating Rate Interest	Interest rate is linked to an external measure and can change throughout the term of the loan	NA		X		X
Compounded Interest	Interest is charged on outstanding interest owed	NA				X
Collection of Interest	Interest is collected with each repayment, or interest is collected at disbursement	NA				

Loan Pricing Options		Our Needs (Ess/Use/NA)	SBMIS™	Jobs!™	Tracker™	MicroTrack™
360-day Payment Year	Calculations are based on a 30-day month	Use		X	X	X
365-day Payment Year	Calculations are based on actual number of days	Ess		X	X	X
Allow Specification of Interest Rate	Interest rates are specified for each loan, rather than fixed at same level for all loans disbursed for that product	Ess				
Interest Calculations Adjusted for Delinquent and Pre-paid Loans	Does the system allow for different interest charges if the client pays early or late?	Use				
Allows Accrued Interest	Interest can be accrued and reported as income before it is paid	Use				
Fixed Up-front Commission	Allows automatic calculation of a fixed amount commission payable prior to or upon loan disbursement	Ess		X		X
Percent Up-front Commission	Allows automatic calculation of a percentage commission payable prior to or upon loan disbursement	Use		X		

Loan Pricing Options		Our Needs (Ess/Use/NA)	SBMIS™	Jobs!™	Tracker™	MicroTrack™
On-going Commission	Allows automatic calculation of commissions on an on-going basis, e.g., monthly	NA				
Allows Over-ride of Interest Charges	User may forgive or reduce interest charges	Ess				
Insurance Charges	Allows automatic periodic calculation of insurance charges	Use		X		X

Funds management

The topic of funds management deals with the way in which disbursements and repayments are handled by the system. Non-profits may have some operational procedures which have not been anticipated by vendors developing software for commercial finance. There are many subtle issues addressed here that can have significant implications for the accounting department.

Loan disbursement form: In what form are disbursements made — cash, check, in-kind, combinations of these, or deposits into client checking or savings accounts? If several forms of disbursement are possible, this has implications for the links into the chart of accounts. A system is needed to indicate which form of disbursement is used for a loan. For example, a loan disbursed by check needs to be credited to the checking account, while a loan disbursed in cash needs to be credited to the teller's cash account.

Payment by cash or check: Similarly different options for loan payments have implications for linkages into the chart of accounts. In addition, payments by check may not clear the bank due to insufficient funds and will need an option to reverse the payment in this instance.

Payment by post-dated check: Some institutions have a policy requiring the client to write post-dated checks for each loan payment when signing the contract. The institution then deposits these checks when each payment comes due. The software can facilitate this process by allowing this series of checks to be entered into the database when the loan is authorized, and then automatically transferring these payments to the loan register when each payment comes due.

Payment by electronic transfer: Loan payments are sometimes arranged as automated electronic transfers. Again, if the system supports this alternative, the procedure can be automated, saving on administrative work.

Payments entered by batch: Rather than entering payments one at a time, accounting departments often prefer to enter payments in batches, or groups. An advantage of such an approach is that the total for the batch can be summed and then verified against the sum of the transactions entered as a means to avoid data-entry errors.

Assign loan to single or multiple fund: Most microenterprise agencies track their portfolios by fund allocation, assigning each client's loan to a particular source of funding. Some systems will allow loans to be assigned to a single fund; others will allow the loan to be apportioned among multiple funds.

Loan disbursement in multiple tranches: Is the loan disbursed in a single transaction or in multiple tranches? Multiple disbursements require more complex programming — for example, to request the amount of the tranche to be disbursed and to compare the previous total plus the new tranche to the approved amount. Not all software packages support this capability. What approval or authorization is needed for disbursement of new tranches? Are there preconditions to disbursements, and if so, how are these verified? A computerized system can aid in verifying that all procedures have been followed, but such functions require more complex software and potentially more customization of the source code. Is the credit product a line of credit, which allows disbursements and repayments at any point as long as the approved amount is not exceeded?

Funds Management		Our Needs (Ess/Use/NA)	SBMIS™	Jobs!™	Tracker™	MicroTrack™
Multiple Option for Form of Loan Disbursement	Can loans be disbursed as cash, check, in-kind, or some combination?	NA				
Payment by Cash	Loan payments can be recorded as cash transactions	Ess		X	X	X
Payment by Check	Loan payments can be recorded as checks	Ess		X	X	X
Payment by Post-dated Check	Allows input of a series of post-dated checks which can then be transferred to the loan payment register	NA				X

Funds Management		Our Needs (Ess/Use/NA)	SBMIS™	Jobs!™	Tracker™	MicroTrack™
Payment by Electronic Transfer	Loan payments can be recorded as electronic transfers	Use				
Payments Entered by Batch	Loan payments are batch processed, with verification of the sum of the transaction	Use		X		X
Assign Loan to Single Fund	Allows fund allocation; a loan can be assigned to a single fund	Ess		X	X	X
Assign Loan to Multiple Funds	Allows fund allocation; a loan can be assigned to multiple funds	Ess		X	X	
Disbursements in Multiple Tranches	Allows the approved loan amount to be disbursed in multiple installments; ideal for a "line of credit" product	Use				

Repayment schedule options

The software will need to be able to calculate the repayment schedule required. It is, therefore, essential to carefully analyze how repayment schedules are prepared.

Term loan with constant payment or constant principal: The most common repayment schedule in the U.S. is an "amortized" system with constant amounts due each period, with the composition of interest and principal varying as the outstanding balance of principal is repaid. Calculations are more complex, but can readily be handled by most portfolio modules. The main alternative is a repayment schedule with set amounts of principal due each period and the interest portion calculated as the outstanding loan balance for the number of days since the last payment. This system results in a total payment which varies from period to period, but the actual cost of the loan is totally transparent to the client.

Irregular payments (set by user): Some institutions require the flexibility to program irregular payments in order to optimally match the cash flow of the client's business. Some software modules will provide a means for the user to manually program the dates and amounts of these payments.

Line of credit: Some loan products allow the client to receive new disbursements upon request up to a pre-approved maximum loan ceiling. The system may not program any required loan payments, allowing the client to make payments whenever he or she desires.

Single payment: Some loan products allow the client to repay the loan in a single payment at the end of the loan term.

Balloon payment: An alternative between a regular amortized loan and a single payment loan is a loan which requires payments at a specified frequency for a period of time and then requires the outstanding principal to be paid in full in a single final payment.

Repayment frequency: What repayment frequencies are allowed for the product? Options include daily, weekly, bi-weekly (every two weeks), semi-monthly (twice a month, not tied to specific week days), monthly, bi-monthly (every two months), quarterly, semi-annually and annually.

Grace period, principal: It is fairly common for institutions to offer a grace period before the first installment of principal is due. Remaining payments are then programmed as regular amounts due at the specified frequency.

Grace period, interest: In some cases the institution also allows a grace period before interest payments are due. Generally, interest is still calculated, but the client is not required to pay the amount until a specified date.

Repayment Schedule Options		Our Needs (Ess/Use/NA)	SBMIS™	Jobs!™	Tracker™	MicroTrack™
Term Loan with Constant Payment	Can create schedule with constant payments where each payment is combination of principal and interest	Use		X	X	X
Term Loan with Constant Principal	Can create schedule with a fixed principal payment and a declining interest payment	Ess		X		X

Repayment Schedule Options		Our Needs (Ess/Use/NA)	SBMIS™	Jobs!™	Tracker™	MicroTrack™
Irregular Payments (set by user)	Allows full customization of the loan repayments	Ess		X		X
Grace Period, Principal	Allows programming of a grace period on principal repayment	Ess		X		X
Grace Period, Interest	Allows programming of a grace period on interest repayment (but no interest calculation)	NA		X		
Line of Credit	Allows a "line of credit" product, which does not have a repayment schedule	Use		X		
Single Payment	Allows a single loan payment at the end of the loan term	Use		X	X	X
Balloon Payment	Allows set periodic loan payments with a large balloon payment at the end of the term	Use		X		
Daily payment frequency	Allows programming of daily payments	NA				X

Repayment Schedule Options		Our Needs (Ess/Use/NA)	SBMIS™	Jobs!™	Tracker™	MicroTrack™
Weekly Payment Frequency	Allows programming of weekly payments	NA				X
Bi-weekly Payment Frequency	Allows programming of payments every two weeks	Use				X
Semi-monthly Payment	Allows programming of payments twice a month	NA				
Monthly Payments	Allows programming of monthly payments	Ess		X	X	X
Bi-monthly Payments	Allows programming of payments every other month	NA				
Quarterly Payments	Allows programming of quarterly payments	Ess		X	X	X
Semi-annual Payments	Allows programming of semi-annual payments	NA		X		X
Annual Payments	Allows programming of annual payments	NA		X		X

Late payment options

Unfortunately not all clients repay their loans on time. This creates yet more complexities and challenges for the portfolio module, which must determine methods for calculation and collection of penalties, optionally allow the institution to reschedule the loan, and in some cases, to write off unrecoverable loans.

Penalty on late payment: Most institutions charge a penalty on overdue loans. However, there are many different methods for doing so, including fixed amount per day, percentage charged to overdue principal and percentage charged to outstanding loan balance. There are also many different rules about when to impose the penalty, e.g., one day after the repayment day or a certain number of business days after the repayment day.

Optional over-ride of penalty: Most institutions will want an option to over-ride the penalty calculations for certain clients. Doing so may require a certain level of authorization and may need to be monitored in an audit trail.

Permit refinancing: Some institutions have a policy to refinance overdue loans. Does the system permit this without issuing a new loan? Are refinanced loans identified as such, as reserve requirements may vary for rescheduled loans. Also an issue is what happens to any outstanding charges, penalties and interest when a loan is rescheduled. Are they capitalized into the new loan principal balance or written off?

Document securities: Some portfolio modules allow the user to input descriptions of all loan security documentation.

Loan write-off: All systems will need some means of writing off unrecoverable loans. As with other sensitive issues, what authorization process is required to write off a loan? Is an audit trail kept? Also, does the portfolio system continue to track written-off loans in an off-balance sheet account to facilitate collection efforts?

Late Payment Options		Our Needs (Ess/Use/NA)	SBMIS™	Jobs!™	Tracker™	MicroTrack™
Penalty on Late Payment	Allows for automatic calculation of late payment penalties			X	X	X
Optional Over-ride of Penalty	Allows user to over-ride late payment penalties			X	X	X
Permit Refinancing	Allows user to refinance loans			X	X	X
Document Securities	Allows input of detailed information on securities offered as collateral			X		X
Loan Write-off	The system allows procedures for writing off unrecoverable loans			X	X	X

7.7. Assess technical assistance, training and contact management modules

7.7.1. Distinction among modules

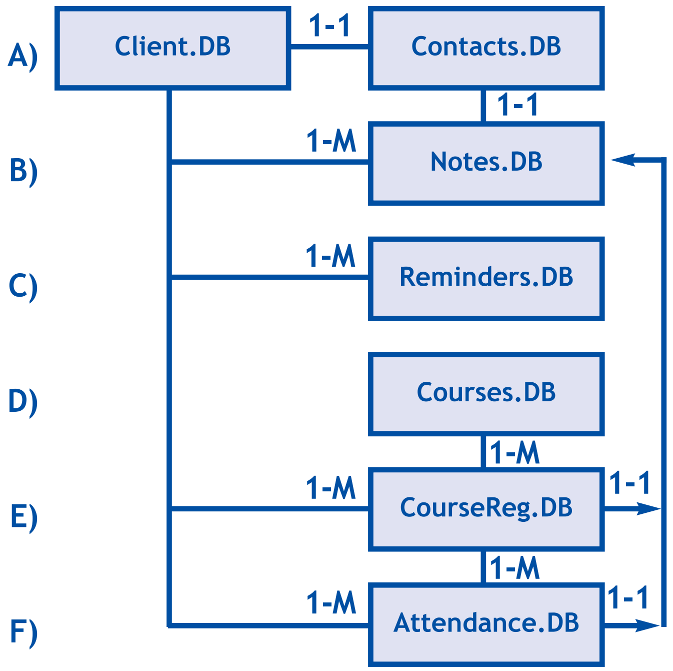
The next three areas — contact management, one-on-one technical assistance and training — have several areas of potential overlap. There is not a clear line of division, and any particular software vendor may combine aspects from more than one of these areas into a single module.

Conceptually, the distinctions are as follows:

- **One-on-one technical assistance:** Any substantive contact (where services are provided between a staff person and an individual client).
- **Training:** Classroom-based participation by the client.
- **Contact management:** Can be interpreted either as "everything else" (what doesn't fit in the above) or "everything" (meaning including what is in the above categories). As "everything else" this module would be used to track incidental contacts, inquiries, correspondence, etc., and to plan future events, such as appointments, renewals and reminders for when required documentation is due.

The distinctions are relevant because they can significantly influence the structure of the tables used to store this information. Contact management modules and one-on-one technical assistance modules can actually be fairly close in structure. Training modules can provide a modest amount of functionality by slight modification of the one-on-one technical assistance module tables. However, for institutions with more ambitious training programs and intensive data needs (e.g., tracking enrollment, attendance, instructor schedules, etc.), table structures would need to be very substantially modified.

The figure below shows an example of how the major tables can be related for these three modules.



Comments

Every individual contact with the client can be registered. Duration, purpose, and staffperson can be noted.

If kept in a separate table, notes can be more flexible than being restricted to one note for one contact. All notes related to the client can be reviewed at once.

Future reminders (request documents) or scheduled appointments can be stored in a separate table.

List of courses by title and date. Not linked directly to clients.

Used to handle registration – who is registered to attend what course. This table provides a many-to-many relationship.

Used to monitor attendance at each session, with one record per participant per session.

The Client.DB and Contacts.DB tables form the core of the one-on-one technical assistance and contact management modules (see Row "A" on page 99). Each contact with the client can be recorded in Contacts.DB, linked to Client.DB via the ClientID key field. The figure below shows a typical listing of fields in the Contacts.DB table. Note that the fields allow statistical tabulation of visits by client, date range, staff person, type of contact or location.

Contacts.DB

Record	ClientID	Date	Staffperson	Duration	Type	Location
1	10500	3/1/2002	CWW	60	Application	Office
2	30220	3/4/2002	SMB	90	Bus Plan	Home
3	10500	3/4/2002	CWW	30	Bus Plan	Business
4	10500	3/6/2002	CWW	60	Marketing	Other
5	30220	3/7/2002	SMB	30	Marketing	Home
6	10501	3/4/2002	AEW	90	Application	Business
7	10501	3/7/2002	AEW	45	Bus Plan	Home
8	10504	3/7/2002	AEW	90	Application	Business
9	10510	3/4/2002	SMB	60	Marketing	Business

Note that Contacts.DB could be used to track technical assistance only, or to track any type of contact, including phone calls, initial inquires, etc. The "Location" and "Type" field drop-downs would need to be modified accordingly. Note that initial inquiries could be tracked only if records were added to the Person.DB table at the point the initial inquiry is made, rather than only at the time the person becomes a "client" of the institution.

In the proposed structure, notes made about the client are stored in a separate table, Notes.DB (see Row "B"). This provides a single place where all comments made to a client's file can be reviewed. Many systems will include notes as a field in one of the tables, but if the system provides multiple modules for technical assistance contacts and training, this can result in notes being scattered around in different tables making a systematic review difficult.

Planning for future events, such as appointments, phone calls and deadlines for submission of documents, is generally handled in a separate table, Reminders.DB (Row "C"), though they can be mixed in with the Contacts.DB information as well. Records are added to Reminders.DB when events or appointments are scheduled. They can be reviewed upon demand, and alarms can be set for each record if desired. After the event or appointment takes place, the record is generally deleted and a new record is created in Contacts.DB to register the information.

Courses.DB (Row "D") would be a fairly small table that contains one record for each course that is scheduled. It would list topic, time, place and instructor. For more complex information management, with multiple instructors, shifting locations, etc., this table would have to be made more complex. It is used in conjunction with the CourseReg.DB table (Row "E") to manage which clients are registered to attend which courses. Because of the structure of this table, each course can have multiple clients registered, and each client can register for multiple courses.

Finally, Attendance.DB (Row "F") can then be used to track attendance by session. Note that both the CourseReg.DB and Attendance.DB tables are linked into Notes.DB, allowing staff to record information relevant to each client, either for the course overall or for an individual session of the course.

7.7.2. Assess one-on-one technical assistance modules

This is a very important module for most microenterprise agencies, as individual technical assistance is nearly universally provided. Because technical assistance is not tracked in companion modules to many of the commonly used portfolio modules, many institutions find they need to purchase independent modules or else develop their own systems.

Track total hours: Total hours can be tracked two ways. The most basic way is a single input cell (field) for each client. The user enters the total amount of training received by the client, but there is no indication of when that training was received. Any time the figure is revised, the previous stored value is over-written and lost. There is much room for input error, and the approach provides no means to track down and correct errors. Although clearly an inferior approach, there are commercial applications that provide this as their only means to track total hours of technical assistance. To provide more detail requires not a single field, but a table dedicated to tracking technical assistance contacts, as shown previously in the Contacts DB table. Total technical assistance hours are then determined by totaling the records for any particular client, staff person or range of dates.

Track individual contacts: This is the approach identified and explained in Row "A" of the figure on page 99. The approach allows each contact to be recorded, along with valuable details about length of contact, type of contact, location of contact and staff person involved.

Track duration per contact: The technical assistance table includes a field to indicate the length of the technical assistance services. Optionally, some systems may include a second field to record transportation time.

Track staff person: The technical assistance table includes a field to identify the staff person providing the technical assistance. Ideally, this staff person is selected from a drop-down list naming all staff members. Note that most systems will allow only a single staff person to be identified for any single technical assistance event. Recording multiple staff would require a more complex table structure.

Identify purpose: The technical assistance table includes a field to identify the purpose of the technical assistance contact. Ideally this is selected from a user-customizable drop-down list, allowing technical assistance to be aggregated by type during reporting and statistical generation. Some systems will include additional fields for sub-purpose for more detail.

User-customizable purpose list: Flexible systems will provide user-customization of the purpose list in order to adapt the list to the organization's services. Selection of purpose from a list enables reporting and statistical generation at a later stage.

Comments per contact: Some systems will include a notes or comments field. This information may be stored directly in the technical assistance table or in a more flexible notes table which is then linked to the technical assistance table. Better systems will offer features limiting future editing or deletion of notes.

Confidentiality: Some systems will allow notes to be marked as confidential, limiting the staff that is allowed to read the information. Some systems will restrict all notes; others will allow the author to mark the status.

Plan future technical assistance: Some systems allow only tracking of actual technical assistance contacts; others will allow input of future contacts into a calendar, or reminder, system. Future visits are frequently tracked in a separate table rather than directly in the technical assistance table.

Resource center tracking: Some applications include a means to track use of their resource centers. This is essentially tracking a "contact" that does not require assignment of a specific staff person. In some systems, it may be possible to assign "resource center" as one of the staff and then register a contact between the client and the resource center to emulate this feature.

One-on-One TA		Our Needs (Ess/Use/NA)	SBMIS™	Jobs!™	Tracker™	MicroTrack™
Track Total Hours	Allows tracking of total hours of one-on-one TA received by client	Ess	X		X	X
Track Individual Contacts	Allows individual record for each TA event	Ess	X		X	X
Track Duration Per Contact	Records duration of each TA event	Ess	X		X	X
Track Staff Person	Records staff person involved in TA event	Ess			X	X
Identify Purpose	Records the purpose of the TA event	Ess	X		X	X
User-customizable Purpose List	Allows user to customize the purpose list	Use	X		X	X
Comments Per Contact	Can record comments for each TA event	Use	X		X	X
Confidentiality	Allows comments to be marked as confidential	Use				
Plan Future TA	Allows planning for future TA, including date and planned purpose	Use			X	X
Resource Center Tracking	Allows tracking the client's independent use of the institution's resource center; time not assigned to any staff person	N/A				X

7.7.3. Assess training modules

As explained previously, training modules can provide a modest amount of functionality by slight modification of the one-on-one technical assistance module tables. In such an approach, a class can be considered a "contact." However, for institutions with more ambitious training programs and intensive data needs, e.g., tracking enrollment, attendance, instructor schedules, etc., table structures would need to be designed specifically to manage this more complex data.

Input by class: This is the more ambitious of the two approaches described above. Tables manage class scheduling, registrations, trainers, etc.

Track class hours: The system allows the tabulation of total classroom hours of training for each client. This provides a valuable measure of services delivered and of services received by the client, which can then be correlated with outcome indicators. Either of the two approaches described earlier have the possibility of tracking class hours per client.

Track attendance/absence: Tracking absences reliably generally requires a class-oriented system, the more complex of the two alternatives. Tracking attendance can generally be done quite efficiently with a class-oriented system, as the enrollment list for the course is presented on a single screen. For client-oriented systems, registering enrollment often means recording "contacts" one-by-one for each client — a more time-intensive process.

Track facilitator scheduling: A class-based system can allow the identification of facilitators for each course, and in some instances for individual class sessions. Such detail is generally only required for organizations managing a large volume of courses.

Class-Based Training		Our Needs (Ess/Use/NA)	SBMIS™	Jobs!™	Tracker™	MicroTrack™
Input by Class	Allows class attendance to be recorded from a class-orientation rather than a client-information	Use				X
Track Class Hours	Allows tracking of number of classroom hours for each client	Ess			X	X
Track Attendance/ Absence	Allows tracking of attendances and absences of each client	Use				
Track Facilitator Scheduling	Allows identification of staff/volunteers involved in teaching class, by session	Use				

7.7.4. Assess contact management modules

A relatively rare feature for dedicated microenterprise assistance software, but one with great potential for using information technology to improve staff productivity and enhance customer service, is the area of contact management. Implementations can be very minimal, such as allowing one short memo field per client per day, to quite extensive, tracking staff person, length and location of the contact, long memos which can optionally be marked confidential, reminders for follow-up contacts, etc.

As explained previously, there is significant potential overlap between a contact management module and a one-on-one technical assistance module. Contact management is much more comprehensive than a technical assistance module. Contact management allows staff to track all contacts with any sort of person or organization, whereas technical assistance modules can often be limited to only specific types of contact with individuals that are considered clients of the institution. Contact management can be used to track discussions and negotiations with potential clients, references of clients, program volunteers, contacts in other organizations, etc. Contact management can also be useful for scheduling appointments and reminders about future contacts and obligations.

There are a number of generic commercial products available to manage contacts and scheduling, generally as part of the PIM, or Personal Information Management, sector of the software market, and sometimes in the dedicated sales management sector. The most well-known examples are Act!,™ Goldmine,™ Maximizer,™ and Microsoft Outlook.™

User-customizable list: Allows the user to customize the list of types of contacts.

Assign staff person: Allows a single staff person to be tied to this contact. Note that most systems will allow only one staff person and one individual per registered contact. Group meetings and such would require a more complex approach.

Memo field: The system allows for notes to be recorded and linked to each contact. Most systems would allow a single note per contact.

More than one memo per day: Some systems allow only a single memo to be created per day, using the date field as the key field. Other systems will allow for more memos and will require another means to differentiate between memos.

Size limit of memo: Size limitations can range from less than 100 characters to unlimited.

Confidentiality: Does the system allow a memo to be marked as confidential and, therefore, restricted to viewing only by the author or other authorized staff? This feature may be essential when dealing with sensitive information.

Can edit earlier memos: Does the system allow the memo to be edited once it is created? Some systems may allow memos to be changed at any time, but this can be abused by staff intending to cover up problems retroactively. Some systems may block the memo immediately after storing it, requiring an additional memo to correct any errors. A nice compromise is a system which allows the memo to be edited and even deleted for a period of time, e.g., one week, after which time the memo cannot be altered.

Can delete earlier memos: Again, for security purposes, the system may prohibit earlier memos from being deleted by the user.

Contact Management		Our Needs (Ess/Use/NA)	SBMIS™	Jobs!™	Tracker™	MicroTrack™
User-customizable List	Allows user to customize the list of types of contacts	Use			X	X
Assign Staff Person	Records staff person involved in the contact	Ess			X	X
Memo Field	Can record comments for each contact	Use			X	X
More than One Memo Per Day	Permits more than one memo to be added with a single date	Use			X	
Size Limit of Memo	Maximum number of characters per memo	>1K			Unlimited	Unlimited
Confidentiality	Allows memos to be marked as confidential	Use				
Can Edit Earlier Memos	Memos can be edited at a later date	Prefer not			X	X
Can delete Earlier Memos	Memos can be deleted at a later date	Prefer not			X	X

7.8. Assess outcomes modules

7.8.1. Keys to outcome tracking

Outcomes tracking — or the ability to evaluate what changes we have made in people's lives as a result of the services they have received from us — is an area of significant concern to most all implementers and donors, but one of the very least-developed areas of commercial MIS software. This is due to two main factors. Firstly, software developers need to know what it is we wish to measure. We clearly specify rules to software companies about how we want to handle our loans — repayment schedules, interest calculations, etc. But, when it comes to outcomes tracking, we haven't yet decided for ourselves what exactly we wish to measure. By this point in the process, you should have addressed this issue during the "Needs Assessment" in Phase I.

After deciding *what* it is you wish to measure, the second challenge in outcomes tracking is determining *how* you plan to collect and manage the data. This again impacts the design of the outcomes module, and there are many serious methodological, procedural and statistical issues to be dealt with.

In the absence of standardization in this area (compare the current status of outcomes indicators to the much more advanced uniformity of loan portfolio management!), there are still useful contributions that software developers can provide. They can provide flexible, open-ended table structures that allow each institution to essentially design its own outcomes tracking system to meet its own specifications. The main approach for developers is to develop systems with two features:

- Outcomes tables that have a one-to-many relationship to the client table, allowing the user to create and store "historical" data. New records can be added to the outcomes table on demand. Each record is identified with a specific client and differentiated by a date field.
- The table should have a large number of user-defined fields with a variety of field formats — most useful are numeric fields and user-customized look-up tables, but text fields and date fields should also be available.

This helps get data into the system, but still leaves a challenge for interpreting the data afterwards. In most cases, advanced data analysis will come not from standardized reports but from directly accessing the data and either running queries or using the data in a statistical package such as SPSS.

7.8.2. Table structure alternatives

Marketing brochures may indicate that a particular software product tracks household income, job creation, etc., but the table structure may not support a thorough analysis if the data is stored simply as additional fields in the basic demographics module rather than in a well-designed outcomes module (see Section 7.8 on outcomes modules).

Demographics modules generally are limited to a single record per client or per business. This means that changes over time — essential for detailed outcomes analysis — cannot be tracked. The key to look for when analyzing a system is if the fields are stored in the equivalent to the Clients.DB and Business.DB tables. If data on income, jobs, etc., is stored directly in these tables, then it is likely that the information is intended only to store values when the client or business begins to receive assistance. Future changes in these values will not be able to be stored in the system without overwriting the initial values. If tracking outcomes is important to your organization, you will need a system with dedicated, independent tables to track the data points that interest you over time.

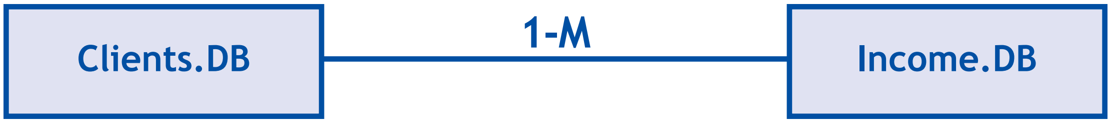
In the first example, income data is not tracked historically, but only as a field in the Clients.DB table. In the second example, where the data is stored in a separate, related table, the information is captured historically, and changes in the client's income level can be analyzed over time.

Example 1: Income stored as a field in the Clients.DB table

Clients.DB

Record	ClientID	Name	Birthdate	Monthly Income
1	10311	John Smith	10/22/1957	$1,500.00
2	10500	Adam Jones	8/29/1963	$1,750.00
3	10501	Cathy Larson	12/8/1975	$2,000.00

Example 2: Income stored as a field in the Clients.DB table



Clients.DB

Record	ClientID	Name	Birthdate
1	10311	John Smith	10/22/1957
2	10500	Adam Jones	8/29/1963
3	10501	Cathy Larson	12/8/1975

Income.DB

Record	ClientID	Date	Income
1	10311	1/15/2000	$1,500
2	10311	11/15/2000	$1,650
3	10500	1/2/2001	$1,750
4	10501	1/4/2001	$2,000
5	10501	6/10/2001	$2,100

7.8.3. Functional matrix

Because of the lack of specific data points to monitor, the feature chart below looks at four main categories that are commonly tracked: household income, job creation, business financials and changes in client perceptions. Typically, each of these areas would be tracked in a separate table, and fields within each table would be customized to match the specific data required by the institution. A key aspect of each area is, of course, the ability to track "historical data" rather than simply a single data point. Those systems tracking only a single data point generally do so by including a field or several fields in another table, such as Client.DB or Business.DB.

Household income level: Can track household income, sometimes by source or family member. The next line in the table indicates whether information can be tracked historically.

Job creation: Can track number of employees, sometimes by different categories such as full-time, or part-time. Some institutions may wish to track characteristics of those hired, such as gender or minority status. The next line in the table indicates whether information can be tracked historically.

Business financials: Allows input of key data points from the balance sheet and income statement of the business. This can be useful for tracking the health of the business and changes in the business over time. The next line in the table indicates whether information can be tracked historically.

Client perceptions: Some institutions may wish to track changes in attitudes, perceptions or behaviors of its clients. This is particularly true of those institutions providing training and technical assistance. Such information is subjective in nature and difficult to collect. A table can accommodate this data if it has a number of user-defined fields, and each field is assigned to a specific question asked in the perception survey.

Outcomes Tracking		Our Needs (Ess/Use/NA)	SBMIS™	Jobs!™	Tracker™	MicroTrack™
Household Income Level	Allows input of household income					
Historical	Data can be tracked historically					
Job Creation	Allows input of job creation data					
Historical	Data can be tracked historically					
Business Financials	Allows input of basic business financial data					
Historical	Data can be tracked historically					
Client Perceptions	Allows input of client perception surveys					
Historical	Data can be tracked historically					

7.9. Assess accounting modules

Thanks to the move toward standard accounting principles, computerizing an accounting system presents relatively few design difficulties. Because of the standardization and the large market for computerized accounting systems, software companies have been willing to create such systems. In addition, the range of choices for microenterprise institutions is relatively extensive, because they do not need an accounting system specific to microfinance or even to commercial banking. Any full-featured standard accounting package is a candidate.

Therefore, this Manual, and the accompanying ***2002 FIELD MIS Software Review*** do not include discussion or reviews of accounting software. The most common issue of concern is the linkage between the accounting system and the loan portfolio module. This issue is addressed in Section 9.2.2.

7.10. Eliminating inappropriate options

The purpose of this exercise is to narrow the selection of viable options down to three to five alternatives. Therefore, this last step concludes by reviewing all the information gathered and eliminating inappropriate alternatives.

7.10.1. Refine and complete the overview comparison

At this point, there is sufficient information to complete the Overview Comparison chart that was first presented in Section 7.1.

Overview Comparison of Software Products

Product->	Our Needs	SBMIS™	Jobs!™	Tracker™	MicroTracker™
	Modules				
Loan Portfolio	2		3	2	3
Demographics	3	3	3	2	2
Contact Mgmt.	2		2		3
One-on-One TA	3	2			3
Training Courses	2	1			2
Outcomes	3	1			2
Accounting	2			Opt	Opt
	Complexity				
Breadth of Services	2	2	1	2	3
Depth of Information	3	2	3	2	2
Scale of Institution	3	1	3	2	3
	Customers				
Software currently used by:		CDFU ACDU	CommVest	MnWorks	ARCLU DcJobs
Decision		Exclude	Possible	Possible	Promising

7.10.2. Screen out inappropriate products

At this point, the team will need to make a decision about which products to eliminate from consideration. Use the "Decision" line on the bottom of the Overview Comparison chart to rate the products on three levels:

- **Exclude:** The product is clearly not a good fit for your institution.
- **Possible:** The product appears to meet your initial screening test and should be researched further.

- **Promising:** Mark a few of the products that look most promising.

In making this decision it is important to look for the right kind of fit between the ratings the team has assigned and the needs of the institution as defined earlier. It is tempting to choose only the systems that appear "advanced," even if the institution's needs have been identified as "basic."

Case Study: Screening out inappropriate products

The task force, with the assistance of their MIS consultant, continued the process by researching the products remaining under consideration after Step 1. They completed the functional matrices and summarized their findings in the Overview Comparison Table above. (Note that there would generally be 10 to 15 products reviewed rather than the four included in this fictitious portion of the case study.)

After careful review, they decided to eliminate SBMIS from further review because of poor fit. Both Jobs! and Tracker were deemed possible solutions to provide for loan portfolio tracking and some demographic data collection. If selected, one of these products would need to be supplemented with a different commercial product or a custom-built system to satisfy their data requirement needs for training, technical assistance and outcomes tracking. MicroTracker was deemed the most promising of the selections, as it provides a complete range of modules, many of which are at approximately the right level of complexity. With all the packages reviewed, there are areas of incompatibility which have been noted. A goal in Step 3 will be to determine the best approach for resolving those incompatibilities.

Chapter 8: Phase II, Step 3: Detailed Software Assessment

8.1. Purpose of the detailed assessment

Those software systems that have survived to this step will now need to undergo a detailed assessment.[27] To this point, you may not have actually run the software on a computer. You may have collected information exclusively through document reviews, correspondence with the vendor and assessments done by third parties (such as the ***2002 FIELD MIS Software Review***). However, you cannot proceed to a final decision without passing through this critical step.

In earlier phases of the review, you will have collected basic information on the system, assembled a checklist of features and made notes about areas that require more investigation. In this phase you will look for any "surprises" that may have escaped your detection so far, and for answers to any questions you still have, such as:

- How does a specific feature really work?
- Is the definition for a certain indicator the same as what we use?
- Do the software calculations perform as expected and reflect rules that are satisfactory to your institution?
- If the module does not perform a certain function, is there a way to adapt the system?
- Can you generate the reports you need?
- Is the overall design and "feel" of the software acceptable?
- Is the performance and responsiveness of the software acceptable?

The tests performed in this final phase will need to provide answers sufficient to make an informed final decision. Everything comes down to one definitive choice after this last look at the alternatives.

The process will not be simple, but it is necessary. On the topic of testing, software vendors and experienced MIS practitioners shared the following comments in a recent discussion on the topic:

- "Although testing is tedious, it is an essential series of steps that helps assure the quality of the eventual system."
- "To test the entire system would be too huge a job, because the number of combinations and possibilities of most systems is simply too large. The main beneficiary would be the software developer (assuming all problems found were reported) and not the MFI."
- "Complete testing of a software package, either by the end user or the software developer is, for all practical purposes, impossible. At first glance, such a statement seems absurd. But, in fact, it is a reality that all software developers and savvy software users understand to be true. There are millions, if not billions, of logic paths in a typical software package. The goal of the responsible software developer is to conduct extensive

[27] Existing documentation to guide users through this critical step is extremely limited. Much of the information for this chapter is drawn from a discussion of the topic that took place on the CGAP Information Services listserve in June 2002.

but not complete testing. As the end user, your goal is to identify a set of functions that are important to your organization and set up a testing process to test those functions."

- "Haphazard, trial-and-error testing will not suffice."

8.2. Preparing for the tests

Understanding and acknowledging the need for a detailed test is not very controversial. However, deciding the degree of effort involved and how to go about the actual test is much more ambiguous.

First of all, the degree of sophistication and complexity of this step depends primarily upon the complexity of your information needs; the more complex your needs, the more thorough the testing required. Those institutions with a wide variety of loan products or complex training programs will need to spend more time insuring that the software adequately handles the variety of products and performs as expected.

Secondly, the less "proven" the software, the greater the need for thorough testing. For commercial software already in use in multiple institutions, this step will not need to be as thorough as the highly-detailed "acceptance testing" required of any custom work you have requested. Any newly created or modified software needs to undergo very rigorous testing to insure its reliability, as inaccurate calculations, misleading reports and lost or damaged data are all quite common with new software.

You certainly will want to view a live demonstration of the software by the vendor. However, do not limit your test simply to a vendor demonstration. Definitely, let the vendor know ahead of time what you are expecting to be shown. Without guidance from you, the vendor will show you just the bells and whistles of the application. A good vendor demonstration should first walk you through all the core functionality you would use on a daily basis. Next it should show the user-defined features and how to set up and edit your financial products, users and other global system level controls. Finally, the vendor can end by showing you functionality you didn't request but might be interested in. All members of your team who will be responsible for making the final choice of solutions should participate in all of the vendor presentations. Depending on the vendor, the functionality and the general price of the product, these presentations can take anywhere from two to six hours, so be prepared to sit for a long while and take good notes.

To prepare for the demonstration and the in-depth testing that will follow, make new copies of the analytical matrices used in Phase II, Step 2 (blank copies can be found in Annex 4). Fill in the first column indicating the requirements for the institution. During the demonstration and testing, fill in the appropriate information as follows:

Column 1: Functionality

Software will perform this functionality either: (1) straight out of the box, (2) with slight modification, (3) with moderate modification or (4) with major modification. Base estimates on discussions with the vendor. Try to get estimates of total programmer hours required for the change.

Column 2: Fit

On a scale of 1 (poor) to 5 (perfect fit), the degree to which the implementation of the feature matches your current operational practices.

Column 3: Comments

If there is a less than ideal fit, describe any discrepancies and note whether it is feasible for the institution to alter its approach (perhaps the software implements a "best practices" approach) or if you think that problems with this feature disqualify the vendor.

Following the vendor demonstration you will need to prepare for your own testing. You will need to decide who will be involved in the tests. Options include:

- You can identify a staff person or two who alone are responsible for all tests. The same people should be involved in all of the tests in order to make accurate comparisons. Regardless of your other options below, you should always include a staff person in the testing.
- Determine to what extent you can or want to involve the vendor in the test. In addition to his or her standard demonstration, the vendor may be willing to go through the software with you, using your sample data, for at least a portion of the testing.
- If you have been using an outside consultant elsewhere in this process, you should continue to involve that person in this stage of the testing. If you have not used a consultant thus far and feel the tests are beyond the abilities of your staff, you should consider involving a consultant at this time.
- Consider using the local university for pro bono services; universities are encouraging their faculty to get involved in the community and providing time for them to do so.

The testing procedures you develop should be fairly consistent among the programs under consideration in order to best compare the results of the tests. Testing should include the interfaces between modules, the correctness of output, and the usefulness and understandability of system documentation and output. The institution should do the following in preparation for the tests:

- Get approval from the vendor to use the system for a certain period of time.
- Get copies of all technical documentation and users manuals that the vendor will agree to provide.
- Review the analysis forms you completed during **Phase II, Step 2: Initial Software Assessment** for ideas on testing. For each specific package you are reviewing, look for any doubts listed during your earlier reviews.
- Develop a list of tests with sample data to be used with each software application. See the following section for a list of suggestions to include in the test.
- If the software package is in use in an institution within reasonable traveling distance, you should visit that installation, observe the system in operation and interview the users about their satisfaction with both the system and the support. If a visit is not possible, users could be contacted by phone or e-mail.

8.3. Performing the test

A standardized test routine should be set up to use with each system. Tests should include all aspects of the program that you anticipate using. The following list serves to start you on the way, but by no means is intended to be complete.

Configuration

- Configure the system for all the different products and services your institution offers.
- Make note of any incompatibilities.

Entering Sample Data

- Set up a small database of about 20 clients and open accounts for them.
- Follow clients through the process of requesting all the different types of products and services.
- Input data for at least three months for these clients. Be sure your test data includes all realistic scenarios. Have some clients drop out or be rejected. For loan portfolio, include pre-payments, pre-termination (early closure of an account), overdue loans, partial payments, etc. For training programs, include perfect attendance, absences, dropouts, etc.
- Carefully test and compare financial calculations of loan portfolio modules. In the words of one experienced practitioner, "The major problems which I have encountered have to do with the factor of time in the calculations. There are a lot of false assumptions in the formulas."
- Attempt to test integration and interface capabilities as much as possible (e.g., try to arrange to have transactions from the demonstration read and transferred to the existing system and vice versa).
- Check out the help functions.

Testing extremes

- Test for extremes, such as what happens if an account is closed the same day or the day after it was opened or disbursed, very large and very small amounts, very long and very short loan terms, interest rate extremes, etc.
- Test the system's edit and validation features with bad data (e.g., Feb. 31).
- Hit incorrect function keys to make it crash.
- Programmers create both valid and invalid test data. These data are then run to see if base routines work and also to catch errors. When testing custom software, created test data should test minimum and maximum values possible as well as all possible variations in format and codes.
- Test unexpected combinations of data, such as events occuring out of the usual order or a client failing a program and later retaking it.

Reporting

- Query all active clients who have been members for more than five years and have children.
- Generate a report of loans in a given geographical region that are more than 30 days late.
- Query all loan officers with clients in arrears.
- Test the process of changing formats for reports, screens and default values.
- Make sure that different reports about the same set of data give consistent results.

Administration

- Set up a new user (employee) on the system with restricted privileges.
- Review transaction logs for errors or security breeches.
- Be sure the software includes functionality to correct end-user data entry errors. For example, can you correct a loan repayment that was credited to the wrong loan account two months ago? As a result, the loan was prepaid when it should have gone overdue, while the other loan has gone in arrears with extra interest and penalty charges.

What you can't always test for

- It is very difficult to test system performance for large databases. If possible, talk with other organizations that use the software and inquire if the system is still responsive when it contains large amounts of data.
- Vendor support is also difficult to determine. In addition to asking the vendor directly, this is something you will need to research by interviewing other institutions that use the software. During these interviews, focus on these questions:
  - What is the response time to serious technical problems (those that make the system unusable)?
  - How successful is the provider in solving technical problems?
- Although desirable in some cases, it can often be very challenging to do "controlled testing," in which the sample data is entered into the old system as well as all new systems under consideration, with the expectation of comparing the outputs. Outputs almost never match precisely, and determining the cause of discrepancies can consume vast amounts of time and energy.

Chapter 9: Phase II, Step 4: Final Decision and System Strategy

The final step in Phase II is now before us: reviewing the information we have collected in order to make a final decision about the software application(s) that best suits our needs. This chapter presents a typology of alternatives, which is then summarized in an "MIS Alternatives Decision Tree" that provides a useful means of prioritizing your choices.

9.1. Overview

There are two groupings in the typology of alternatives: (1) integrated systems versus hybrid systems, and (2) commercial software, modified commercial software and custom software.

The distinction of **integrated versus hybrid** systems is related to whether the various modules required by the system share information seamlessly. Generally, an integrated system has all its modules provided by the same company, whereas a hybrid system is a collection of different modules from different vendors that generally suffer from some duplication of data, some incompatibility of definitions and difficulty in aggregating information for reporting purposes.

The second key aspect is **commercial, modified commercial or custom** software. Commercial means the software is used essentially "off-the-shelf," i.e., it has the same design and features as used by other clients of the software company. Modified commercial software means that the client has paid the vendor to make some custom adaptations to the standard software in order to make the software better meet the needs of the institution. Custom software means that the client has hired a company (or hired internal staff) to develop software from "scratch," specific to the exact needs of the institution.

The following sections provide more detail on these different options.

9.2. Integrated versus hybrid systems

The first distinction to understand is that between integrated and hybrid systems, or whether various modules required by the system share information seamlessly (see Section 2.3.1 for more information on modules and issues of integration). The ideal that all institutions strive for is to have an integrated MIS in which all information managed by the institution is interconnected and related.

If the system is well designed, there is no duplication of data entry and all information on activities can be easily correlated and analyzed. However, as emphasized repeatedly throughout this Manual, such a high degree of integration is very challenging and quite rare. Our estimates are that less than 20 percent of microenterprise agencies operate with an integrated MIS. The vast majority, therefore, have "hybrid" systems, in which discrete systems are used to track different aspects of the institution. An example provided earlier in the Manual was that of West Company, which uses Quickbooks Pro™ for its accounting, Down Home Loan Manager™ for its loan portfolio management, MIS IDA for its IDA services, and a custom Microsoft Access™ database to manage its clients and training services.

Living with a hybrid system is a fact of life for most institutions and the situation is not likely to change for the majority. Instead of continually looking to change software, the goal for these institutions should be to focus on improving the way their modules interface with one another.

9.2.1. Partially integrated systems

Be forewarned: Systems that are marketed as "integrated" can still have less-than-ideal integration of the actual data. Integrated systems are generally supplied by the same vendor. However, the different modules may have been developed by different programmers or at different times, resulting in design or user-interface inconsistencies. Or, the vendor may have purchased modules from another company and integrated them into his or her product, again resulting in potential inconsistencies. Alternately, vendors may advertise that they have provided means to transfer data automatically from one module to another (e.g., transferring loan data to a popular accounting package), but in actual practice this integration may be far from optimal. In practice, the integration may work no better than a custom integration procedure that your organization develops to integrate your hybrid system.

9.2.2. Developing links in hybrid systems

Integrated systems are generally comprised of a collection of modules developed by the same company to the same user-interface standards and internal data definitions. Data for each module is stored in tables within each module and these tables are then related to each other in the ways described in Section 2.3.2 of this Manual. If the modules are well integrated, the data in these tables can be related seamlessly. For example, you may select a particular client and get a listing of all data related to that client — loans, training, contacts, contact history, etc.

However, if the data is stored in individual modules that have not been fully integrated via the user interface, extracting comprehensive information can be a chore sometimes bordering on the impossible. To assemble the client information from the previous example might require separately opening three or four different software applications, searching for the client by name (or client number, which would differ in each database), locating the information in question and then recording the results manually in a summary document.

In some cases, linkage routines that your organization creates to integrate modules and develop reports may work as well as those marketed commercially. Much depends on the database storage format and whether the vendor allows you to access the tables from outside of the system. While such accessibility has improved with the use of ODBC drivers incorporated into the Windows™ environment, newer database storage formats are often password-protected and you will need to ask the vendor to provide the necessary passwords.

If the tables from the various modules can be accessed externally, it is possible to write an application using a program such as Microsoft Access™ to open all the relevant tables and extract the relevant information from each one to create a summary report, or to transfer some specific information over from one module to the second module, perhaps storing the data in some of the user-defined fields of the second module. In some cases, it may be necessary to create new tables for use by the linkage routine, e.g., a "look-up" table to match the Client IDs for the two systems, as these identifications will rarely be identical.

Linking accounting and portfolio systems

Transferring information back and forth between the accounting system and the loan portfolio system is a special situation, because both systems track financial data and that data must fully coincide. It is a common belief that the computerized portfolio system (which tracks individual client accounts) and the computerized accounting system (which tracks activity at a more aggregate level) must be seamlessly linked so that all transactions entered in the portfolio system are automatically reflected in the accounting system. In reality, this is not necessary, as careful accounting policies can insure that activity posted in each module is consistent and all account balances remain in sync.

Client loan accounts and the accounting system are actually “linked” through accounts on the general ledger (see example box for more details). The total loan balances in the client accounts must match the balances in the corresponding general ledger accounts. Because the client account balances total to match the general ledger balances, the client accounts and the general ledger should be periodically reconciled (at least monthly in a small institution and daily in a large, regulated financial institution) to insure that proper information is being recorded in both.

Using the Chart of Accounts to control loan activity

The chart of accounts is the key to linking portfolio data with the accounting system. The chart of accounts should have a different account for each loan product, funding source or branch office, whatever level of detail you are interested in tracking. This allows you to consult the loan activity and outstanding balances for each characteristic of interest. For example, assume an institution with two loan products:

1210	**Loan Product A**
1220	**Loan Product B**

Each time there is a new loan transaction for a given product, either a disbursement, repayment or write-off, the account number will link it to the appropriate category on the balance sheet. In a portfolio system which is not linked automatically to the accounting system, the aggregate totals can be transferred periodically. For example, all new disbursements for Loan Product A will be totaled and entered as a Debit to account 1210. All loan principal repayments will be totaled and entered as a Credit to account 1210. The balance for account 1210 will maintain the proper total balance of outstanding loans for Loan Product A at any given point in time.

Likewise, any interest, fees and penalties will be tracked and entered as Credits to the respective income accounts, e.g., Account 4100 (Interest), 4200 (Fees) and 4300 (Penalties).

The most common error in this transfer process is when aggregate totals have already been transferred to the accounting system and then an error is identified. If the error is corrected only in the portfolio system and not in the accounting system, the balances will no longer coincide. Any adjustment made to one system must be made to the other. It is always best practice to enter adjustments as a separate entry on the day on which they are identified rather than to go back and change historic data. Tests can be performed at any time by using the portfolio system to run a report for the outstanding balance for each loan product and comparing the figure to the balance for the corresponding account number in the chart of accounts.

In fact, institutions can actually be better off not linking client accounts and the general ledger by computer. A non-linked system provides another level of internal control, offers more user flexibility and less computer dependence, and is less expensive because it does not demand additional programming or software support. The accounts could run on a separate, inexpensive accounting package and the client accounts on a programmed database. A transaction report could be printed from the client accounts program detailing the loans disbursed, loan payments received, journal vouchers adjusted and savings deposited. The totals and the individual transactions could then be reconciled with the accounting transactions and the original paper slips. These procedures could be performed daily for greater reliability of client account data, with any irregularities identified and resolved on the day they occur.

9.3. Commercial systems

Commercial means the software is used "off-the-shelf," i.e., it must have exactly the same design and features built into the software as used by other clients of the software company. However, this does not mean that the software will perform exactly as with other clients. Software developers can build in various configuration options so that the software behaves differently for different clients.

Configurability can be done either through user-accessible screens (e.g., choosing from a list of options for calculating interest) or by vendors using undocumented procedures (e.g., setting "switches" to enable different features, for which the vendor may charge an additional fee). If you need a feature that doesn't appear to be available in the software, always ask the vendor, as that feature may be available, though undocumented.

A high degree of configurability is generally the sign of a mature software product that has been in development for a reasonable amount of time and is in use in a large number of institutions. Generally, increasing the client base means the developer encounters more diversity in operational procedures and builds in responses to that diversity in the form of configurable options in the software.

Interestingly, there is a number of software products being sold that have actually been in development "too long." What this means is that the software was developed using what were good development tools at the time, but those tools are now outdated. The software language and the database engine may now be out of date, the product discontinued or the software company no longer in existence. The dilemma for developers is that their efforts are often not transferable from one development platform to another — to upgrade their software to a new environment often means completely starting over. Thus, it is not that uncommon to find software vendors trapped continuing to market their older applications.

9.4. Modified commercial systems

Many software packages are available in two forms: an "off-the-shelf" version, in which customization is strictly limited to configuration options available through the software, and a custom version, in which the software firm either: (1) incorporates additional modules and routines not in the standard version, (2) modifies routines to include new features, or (3) writes new routines to the client's specifications.

Clients must be forewarned not to assume that any change, regardless of how inconsequential it seems, can be minor or low-cost. Custom programming is generally charged at rates well in excess of $100 per hour, and even small changes often take days to implement and test. In addition to the high cost, the client needs to be aware of several technical issues that can create problems.

Firstly, source code for a complex MIS can often be modified by only a handful of people — ideally, the original programmers. Source code is complex and often poorly documented. Even with good documentation, it is difficult for another programmer to fully understand how the source code operates. Any change, no matter how minor, needs to be carefully tested and debugged, because a change in one area of the program can have unexpected consequences in apparently unrelated areas.

Secondly, customization causes potential problems for the software firm in future upgrades. If a change for one institution is incompatible with the installation in another, the software firm must maintain multiple sets of source code for future improvements, and upgrades — or even bug fixes — can become a nightmare.

All of this often results in high costs for customization. In fact, a few small changes in the software can often significantly exceed the cost of the basic "off-the-shelf" program. In summary, customization of a solid software package to meet an institution's needs is often a good alternative. But, given the cost and the potential for bugs, such customization should almost always be limited to the essentials and should be carefully thought out beforehand.

9.5. Custom systems

After reviewing commercial software options, many institutions have chosen to develop either a custom integrated system or a custom module to complement their commercial software modules. They express several reasons for their decision: lack of solid, identifiable alternatives for purchase, lack of a fully integrated system, a preference for a system fully compatible with their operations, and concern about ability to improve and modify the system to meet future needs.

These are legitimate concerns, particularly if an institution is considering a small custom module to complement its other software, e.g., a module to track training course attendance. However, if an institution is considering development of a fairly large system or an integrated system to avoid having to manage multiple, incompatible MIS modules, they should proceed with extreme caution. The experience of the vast majority of institutions that have chosen this route is that the results are far from their original expectations. Development of a small, basic application[28] in Excel™ or Access™ can sometimes be done even without the aid of experienced programmers. However, developing a more complex application, such as a loan portfolio module or a fully integrated system is far more challenging. In a disturbing number of cases, the software simply fails to work and the effort is abandoned. We would not be acting responsibly if we did not strongly caution you about pursuing this option, as tempting as it may initially seem.

What can go wrong? Everything! In virtually all cases, the software development takes far, far longer than planned, generally two to three times longer. Costs also vastly exceed original estimates as the development period drags on.

[28] Such a system would meet the needs of an institution with a score of 3 or 4 out of the maximum 9 points on the "dimensions of complexity" framework introduced in Section 2.3.

Whereas full functionality is often a primary motivator in choosing this route, much of the planned functionality never makes it into the custom system. Instead, energy is directed into finalizing and debugging core, or basic, functionality. As the budget dwindles, the additional features get postponed. Systems get implemented once the core features are in place. Attention then turns to debugging, a process which can in itself be quite time consuming as well as frustrating for operational staff, which are relying on an "unreliable" system to give them accurate information. Reports are often the last aspect of the system to be developed. In the meantime, staff members learn to manually generate the reports they need.

In most cases, software vendors have invested years of effort in developing and refining their products, at great cost. That cost is then distributed over a (hopefully) large user base. The result should be a better system at a lower cost than what an institution can develop on its own. However, many times managers look at existing systems and think they can do better. They talk to programmers that claim it wouldn't be that hard to develop a system. Initial time and cost estimates come in looking very promising. But, things become more complex as the development process gets underway, for many of the same reasons cited throughout this Manual, including: poor definition of information needs, lack of clear and consistent operational procedures and policies, an inability to recognize how many exceptions there are to the norm, and not understanding how difficult it is to develop software that adequately handles those many exceptions.

One contributing factor is that programmers often are inexperienced in the specifics of the microenterprise institution's business. They may be developing a loan portfolio system without really understanding how interest is calculated and how delinquency is measured. Or, they assume that since they have written an inventory control system for an auto parts store, they can easily manage a class scheduling system. Often they want to leap immediately into system development. But, if the system is not well conceived beforehand, the development can take much longer with major elements of the system needing to be reworked. And, in the worst cases (unfortunately, not all that uncommon) the system may never work properly. The importance of following a systematic process of needs assessment and system design cannot be overemphasized.

Some institutions prefer custom systems in order to have control of the process and the final product. A custom system can be developed in-house by staff of the institution, which insures access to the source code and provision of technical support, although the ongoing costs of that support may be high. Or, development can be contracted out to an independent firm, in which case ownership of the source code and the cost and reliability of technical support need to be carefully negotiated. With control of the source code, the institution has much more control over future alterations and additions, or critical bug fixes.

Despite all of the dire warnings presented thus far in this section, we need to include one final strong cautionary note before closing. In our research, we heard plans from a significant number of institutions that were developing or had developed custom software at significant expense, in the hopes of being able to then market that software to other institutions and recoup some of their investment. In only one case that we know of did this actually work as intended. In all other cases, the institutions decided that the software would require significant additional investment before it would be worthy of distribution to other institutions. In addition, many also commented that they realized their strength and expertise were in microenterprise assistance — not in software development and support — and that any attempt to add software marketing to their institution would distract from their core business. Several institutions were considering fall-back plans of selling their custom software to an existing software company, so that that company could expand upon their work and make it available to others.

9.5.1. Fundamental principles of good database design

Case Study: Staff members learn the fundamentals

As will be explained in the next chapter, Women's Initiative decided to design and develop a custom database to monitor their complex range of training and technical assistance services and track the data necessary to implement their outcomes monitoring initiative. As they undertook the design of a custom database, they found it extremely helpful for all of their operational staff involved in the design process to understand some fundamental principles underlying good database design. They developed a list of guiding principles that were readily understood by staff, resulting in significantly enhanced dialogue between operations staff and their database developer.

This list appears as the remaining portion of this section.[29]

Good database design is typically anchored in creating a good set of tables, where a table is simply a two-dimensional matrix of data with a title (name), rows (records) and columns (fields). Each row is a line of data. Each column is a field containing some kind of information. For example, an "employee" table may contain four fields or columns: "ID," "Name," "Department" and "Salary." Each row of the table would contain information about one employee. For example:

Employee Table

EmployeeID	EmpName	Department	EmpSalary
100	Betty Smith	Marketing	48,000
140	Allen Beeton	Accounting	52,000
110	Chris Lucerno	Info Systems	43,000

Designing tables may seem simple on the surface, but in practice can be quite complex. It is arguably the top reason that a particular software application may or may not suit your needs. Following are some guidelines for creating good database structure:

- A database is comprised of a collection of tables, each with a unique name.
- Each table describes only one category of data, or "entity," for which an organization wishes to collect data, e.g., employee, customer, training course.
- Tables are comprised of columns (fields), each with a unique name.

[29] The list of principles provided in this section is based on the original list developed by Women's Initiative but modified by Tim Shea and Chuck Waterfield.

- Each table must contain a "key field" that will be unique for that row (record), e.g., CustomerID, EmployeeID, CourseID.
- All other fields in the record provide additional information to the unique ID (the key field). For example, the Name of Employee 100 is "Betty Smith" and she works in Marketing.
- Each "cell" (the intersection of a row and column) can contain only a single value. For example, in the sample table above Employee 100 can be listed only under a single name, Betty Smith, and may not have an alternate name listed, e.g., Betty Jones-Smith, in the same cell.
- All the non-key fields are "mutually independent," i.e., there are no calculated results stored in the table, and you should be able to update any column without impacting any other column or row.
- The key fields are also used to relate the tables to each other in a one-to-one or one-to-many relation. For example, one client can take many courses: one client record (key field ClientID) can relate to many course records (key field CourseID). So in the Courses Table include a field for ClientID (a "foreign key"). That way when you've identified a client, you can also identify all the courses the client took.
- Data that is input must follow "business rules." Business rules establish what type of information can be input into a field, e.g., a numerical value less than 99,999 with no decimal places, as well as particular relationships between a specific pair of tables, such as one-to-one relation or one-to-many relation.
- Data that changes through time requires special storage designs if you wish to retain the historical information. Each record must contain a date field, and careful consideration must be given to defining the content of the date field consistently in all cases.

9.6. Contracting issues

When contracting a developer or firm to develop custom software, the microenterprise agency will be responsible for developing the contract. When purchasing commercial software, contracts are generally provided by the vendor. In either case, both parties have every right to request changes in the contract before signing.

When reviewing a contract provided by the vendor, you will need to read through everything very carefully, as in many cases the standard contract will tend to protect the interests of the software vendor more than those of your institution. Be sure that any specific agreements about features and changes have been properly incorporated. Don't be afraid to ask for any modifications to the contract, and don't sign any contract that contains clauses you are uncomfortable with.

Seek to achieve the following objectives with any contract:[30]

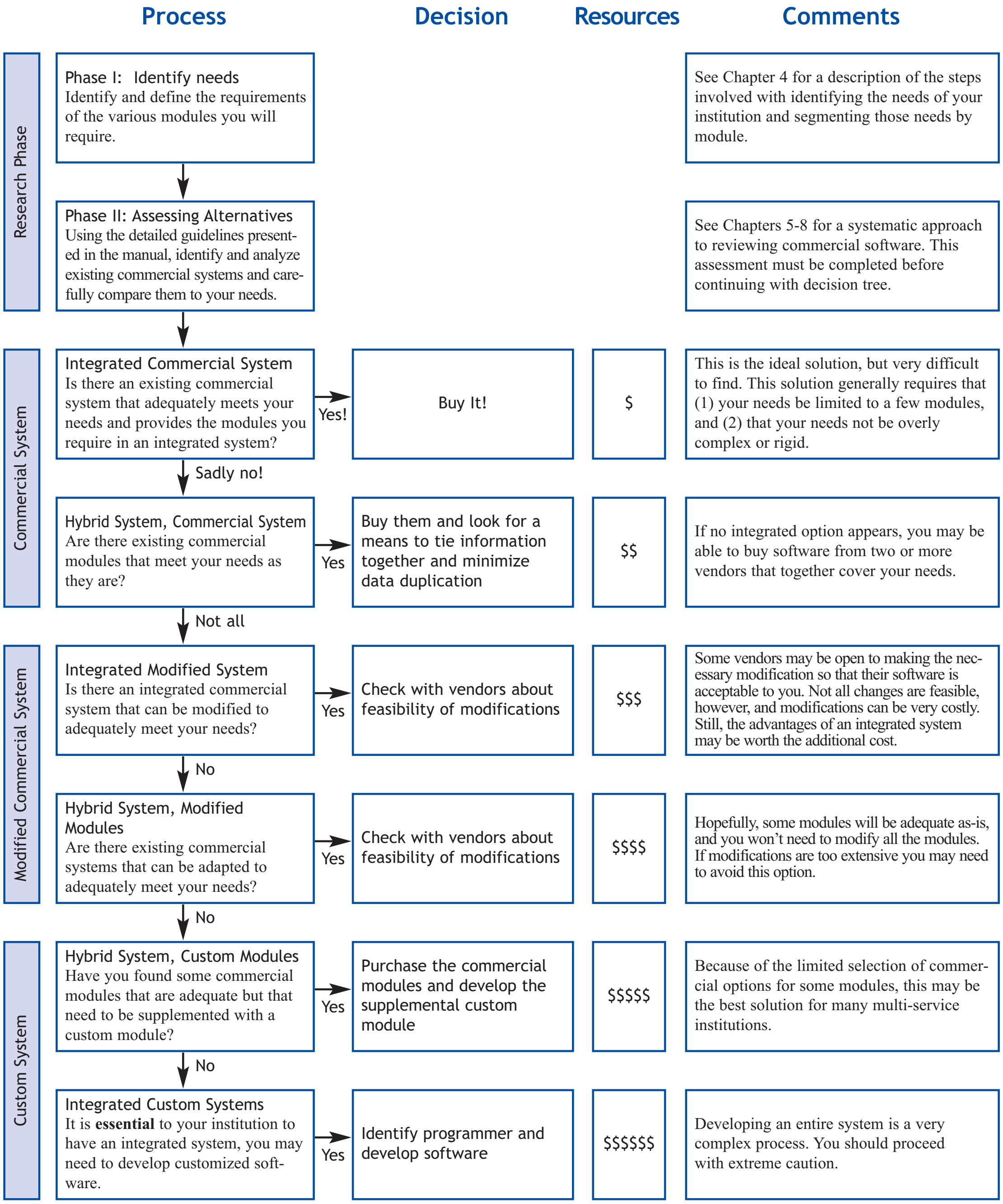
- Clearly identify your expectations as a buyer to avoid misunderstandings.
- Identify recourse should the vendor fail to comply with the requirements of the contract.
- State a contingency plan should the vendor go out of business, e.g., indicate that the vendor will provide a copy of all source code and system documentation at no additional cost.
- Be sure to have an attorney knowledgeable in software contracts review the document.
- Be wary of accepting the vendor's standard contract, as it generally protects the vendor and not the buyer.
- Put everything in writing!
- "Negotiate in good faith and do not make unreasonable demands."

[30] David Murphy, *Microfinance Information Systems Selection* (unpublished manuscript).

9.7. Negotiating with vendors

- Don’t just listen to the sales pitch.
- Vendors aren’t lying, they just aren’t objective; they see things from their perspective and their product seems right.
- Bring in an outside consultant for a few hours to listen in on discussions; local consultants have more vested interest; stay away from big firms that send a different person every time.
- Request changes that will be marketable beyond your institution.

MIS Alternative Decision Tree



	Process		Decision	Resources	Comments
Research Phase	Phase I: Identify needs Identify and define the requirements of the various modules you will require.				See Chapter 4 for a description of the steps involved with identifying the needs of your institution and segmenting those needs by module.
	Phase II: Assessing Alternatives Using the detailed guidelines presented in the manual, identify and analyze existing commercial systems and carefully compare them to your needs.				See Chapters 5-8 for a systematic approach to reviewing commercial software. This assessment must be completed before continuing with decision tree.
Commercial System	Integrated Commercial System Is there an existing commercial system that adequately meets your needs and provides the modules you require in an integrated system?	Yes!	Buy It!	$	This is the ideal solution, but very difficult to find. This solution generally requires that (1) your needs be limited to a few modules, and (2) that your needs not be overly complex or rigid.
	Sadly no!				
	Hybrid System, Commercial System Are there existing commercial modules that meet your needs as they are?	Yes	Buy them and look for a means to tie information together and minimize data duplication	$$	If no integrated option appears, you may be able to buy software from two or more vendors that together cover your needs.
	Not all				
Modified Commercial System	Integrated Modified System Is there an integrated commercial system that can be modified to adequately meet your needs?	Yes	Check with vendors about feasibility of modifications	$$$	Some vendors may be open to making the necessary modification so that their software is acceptable to you. Not all changes are feasible, however, and modifications can be very costly. Still, the advantages of an integrated system may be worth the additional cost.
	No				
	Hybrid System, Modified Modules Are there existing commercial systems that can be adapted to adequately meet your needs?	Yes	Check with vendors about feasibility of modifications	$$$$	Hopefully, some modules will be adequate as-is, and you won’t need to modify all the modules. If modifications are too extensive you may need to avoid this option.
	No				
Custom System	Hybrid System, Custom Modules Have you found some commercial modules that are adequate but that need to be supplemented with a custom module?	Yes	Purchase the commercial modules and develop the supplemental custom module	$$$$$	Because of the limited selection of commercial options for some modules, this may be the best solution for many multi-service institutions.
	No				
	Integrated Custom Systems It is **essential** to your institution to have an integrated system, you may need to develop customized software.	Yes	Identify programmer and develop software	$$$$$$	Developing an entire system is a very complex process. You should proceed with extreme caution.

9.8. MIS alternatives decision tree

The previous graphic depicts and describes the various alternatives to be considered. The choices are presented in decreasing order of preference, based on complexity and cost issues. The tree appears complex at first, but becomes clear once the distinctions among the main categories are understood: (1) integrated systems versus hybrid systems and (2) commercial software, modified commercial software and custom software.

The top portion of the tree summarizes the research phase that has already been detailed in earlier chapters. After collecting all information, the institution needs to reach an appropriate decision, and the alternatives can be listed in descending priority as follows:

Integrated Commercial System: For most situations, this would be considered the ideal solution, but very difficult to find. Generally, this alternative implies that your needs are limited to a few modules and are not overly complex or rigid.

Hybrid Commercial Modules: If no integrated option appears, you may be able to buy software from two or more vendors that together cover your needs.

Integrated Modified System: Some vendors may be open to making the necessary modifications so that their software is acceptable to you. Not all changes are feasible, however, and modifications can be very costly. Still, the advantages of an integrated system may be worth the additional cost.

Hybrid System, Modified Modules: You may find you need to assemble a system from various vendors. Hopefully, some modules will be adequate “as-is,” and you won’t need to modify all the modules. However, if modifications are too extensive, you may need to avoid this option as it can be both complicated and costly to request modifications to more than one module.

Hybrid System, Custom Modules: In this instance, there are commercial modules available that meet most of the needs of the institution, but together the modules do not cover all of the needs of the institution and must be supplemented with a custom module to fill the gaps. Because of the limited selection of commercial options for some modules, this may be the best solution for many multi-service institutions.

Integrated Custom System: Developing an entire system is a very complex process. You should proceed with extreme caution. Refer to issues in section 9.5.

Three main questions driving the choice among these alternatives are:

- **How much money is the institution willing to invest?** Increased sophistication, functionality and customization generally result in higher cost.
- **How flexible is the institution willing to be in adapting its policies and procedures to the system under consideration?** Though difficult to accept, the fit between the software and the institution will never be perfect. And the software, while currently functional, will always have a lengthy list of additional ideas and features to incorporate into future versions. Be content with having the MIS meet 80 to 90 percent of the institution’s needs and wants. The more demanded of the MIS, the more complex it becomes and the less likely that it will operate trouble-free.

- **How reliable is technical support for the system under consideration?** Systems can crash for any number of reasons, including operating systems, user error and software conflicts, as well as actual bugs in the software application itself. Regardless of culpability, it usually takes technical expertise to get the application running properly again. If technical support is not timely or reliable, an institution could be forced to operate without a functioning system — an unacceptable proposition for institutions highly dependent upon their MIS.

All three of these questions need to be kept in mind, balancing the aspirations of the institution with the practicalities of the alternatives before them when choosing among the three options.

Case Study: Reaching a Final Decision

After a careful analysis of their information needs and a review of existing software, Women's Initiative ultimately chose a strategy of combining an array of modules, some commercial and some custom software developed for their specific needs. They addressed their main accounting needs with a full-featured accounting program, Financial Edge™, designed for non-profits, which links with Razor's Edge™, used to track resource development, volunteer information, and organizational contacts. They chose a specialized, commercial package, Down Home Loan Manager™, to manage their loan portfolio. And, to manage demographic data, training and technical assistance service delivery, and outcome measurement, they decided to design their own specialized system. They located an experienced programmer who was willing to commit to the effort as a volunteer. Together, they went through a thorough needs assessment, designed the structure of the system and decided to continue with Microsoft Access™ as the platform. They found the process to be more complex than they originally anticipated and to require their staff's active engagement from start to finish. Yet, the benefit of this level of effort by Women's Initiative is a much better MIS than they would have otherwise located or created.

As a result, Women's Initiative found a very suitable option to meet their complex and advanced information needs by blending together three different software applications from three sources. Other institutions will find other solutions, depending on their specific situation and their resources and priorities.

Chapter 10: Phase III: System Implementation

10.1. Overview

The following figure summarizes the steps to be followed during Phase III: System Implementation. These steps are described in detail in the remainder of this chapter.[31]

Phase III: System Implementation

System Implementation:

- Setting up the Hardware
- Preparing and revising documentation
- Configuring the system
- Testing
- Training
- Transfering the data
- Choosing Plunge, Pilot, Phase or Parallel
- Data back-up Policies
- System Maintenance

10.2. Selecting and setting up the hardware

People often obsess on the software aspect of MIS and overlook the importance — and the cost — of the hardware necessary to run the new software. Setting up the computer hardware for a new system can be time-consuming, and it requires much anticipatory planning, especially in purchasing decisions. A new MIS may require investments in new computers, printers, power supplies, back-up units, operating systems, additional software, cabling and other peripherals.

In addition, new hardware may require additional investment in office infrastructure, with phone lines, Internet access, network cabling, remodeling of work areas or secure storage of the network server.

10.2.1. Tips on selecting hardware

NOTE: This section contains generalized advice on purchasing hardware.[32] **Recommendations on specific components are included in a lengthier treatment of this topic included in Annex 5.** Given the rapid changes in computer technology and pricing, that information has been separated from the body of this Manual in order to allow it to be updated as current technology standards and pricing change.

Settle on one vendor. Try to stick with one vendor for all your computer purchases as much as possible. One vendor sourcing can ease a small organization's maintenance procedures. You won't have to remember who to call for computer troubles when different systems encounter troubles. This should allow you to reduce overall information technology costs and maintenance.

Compare systems even within the same vendor. If you are careful and keep an eye out you can for the most part get a higher-end system for a little more cost than a mainstream

[31] This chapter draws extensively from material in the *CGAP MIS Handbook,* "Chapter 5, Step 4: System Implementation."

[32] This section was written by Todd Schuldt and Tim Shea, members of the FIELD MIS Advisory Task Force.

system. Most vendors such as Dell and Gateway add components to their higher-end systems such as DVD's, high-end video cards and CD-RW's that you don't necessarily need in a typical workstation.

What size "box"? Those micro-towers and micro-desktops may look cute but expansion to that system will be limited. If you later decide to add a CD burner (CD-RW), you may have to remove the existing CD-ROM in order to do that (check how many "bays" the unit contains). Also, the motherboard may only have two or three expansion slots, so adding internal devices such as a SCSI controller, network card or modem may be limited. Also your upgrade avenues such as motherboard and/or processor replacements may be more restrictive.

Should I get the extended warranty? Unless you have a full-time information technology person that is comfortable in working with hardware, buy that three-year **on-site** warranty. There are several types of extended warranties available ranging from five-hour to three-day response times to on-site or carry-in service. Usually you can find a three-year on-site extended warranty from major vendors for as little as $99. What this type of warranty means to you for the most part is that for the next three years if any hardware malfunctions on that system, the vendor will send someone to your office the next business day with the replacement part and install it at no charge to you. Talk with your vendor about the price differences among the warranties and what they could mean for you.

Buy now or buy later? Look for the best deal that you can find at the time you need the new computer; make your decision and move on. The volatility of the PC market is such that in four weeks a different computer system may be available for less, but when you need that computer today, that will not help you. With the right shopping you are going to purchase a system that can be upgraded later anyway if need be.

Buy for the long term. Unless you are a for-profit company, you are most likely going to be using that computer workstation until the day it literally dies. By spending just a bit more for an upgradeable computer, you'll be able to keep that computer alive for much longer than if you try to save a little bit during the initial purchase. Computer advances are leveling off; if you buy a good unit, rather than a unit that is already reaching the end of its useful life, you should be satisfied with your purchase for years to come — but also avoid buying the "leading edge" computer, as the additional cost for the very latest in technology is rarely justifiable.

What about leasing? Alternatively, consider leasing your computer equipment. There are two advantages. Firstly, there is less up-front cost. Secondly, it avoids the organizational frenzy every three or four years when you need to justify a big purchase of new computers. Lease payments become a standard operating expense not very different from a monthly phone bill. Keep in mind though that you will need to overlap lease agreements slightly to minimize the disruptions of changing over almost all of your systems at once at lease end. If possible, stagger your leasing agreements so that no more than 50 percent of your systems' leases are up at the same time.

10.3. Preparing and revising documentation

The institution's documentation on policies and procedures will need to be revised to reflect any procedural changes introduced in conjunction with the new system. Good documentation can be invaluable in ensuring proper use of the system, especially in large, decentralized organizations or in organizations undergoing expansion. It can also serve as a training tool for new staff and assist staff in dealing appropriately with new situations.

10.4. Configuring the system

Most software installed in more than one institution uses configuration options to set up the system for an institution's needs. Configuration options are generally menu-driven and accessible by a user registered at the level of system administrator. Less commonly used configuration options are enabled by special codes entered into a configuration file by a technician familiar with the software.

Configuration consists primarily of:

- Setting up the structure of the chart of accounts. This crucial task may require modifying the institution's chart of accounts to match the operations of the software package.
- For portfolio packages, defining the financial products, each with myriad rules — such as minimum and maximum amounts, interest calculation methods, links between accounts and treatment of delinquency for loans. If the software is sophisticated, the list of options can be surprisingly long.
- Establishing numbering conventions for clients and for loan accounts and IDAs.
- Establishing relationships among branch offices — for example, for sharing and consolidating information.

10.5. Testing

The next step is to test the system with actual data, if not already done as part of the evaluation process. Historical information for the past several months on 50 to 100 clients should be entered into the system.

This testing phase serves two purposes. Firstly, it allows the development of a strategy for data conversion or entry of initial data for all active accounts. Secondly, it allows careful study of the system's behavior:

- Are staff using the system properly, e.g., entering data according to the rules?
- Has the system been properly configured, with all customizable lists preloaded with the necessary values?
- Are calculations being done properly, especially with respect to loan portfolio modules?
- Do reports contain correct information?
- Does the system crash inexplicably?
- Does the network function adequately?
- Does the system allow correction of data entered in error?
- Is the system user-friendly, or are there urgent issues that need to be addressed?

10.6. Transferring the data

Data transfer is one of the biggest unknowns in a MIS installation. It requires careful and deliberate decisions and guidance, preferably from an expert who has been through this minefield before. When installing commercial software, it is best to obtain advice from a technician familiar with the system. Where large quantities of data are involved, there is great potential for disaster — the wrong decision can mean weeks of lost time because data need to be re-keyed, or months of frustration because data in the system bear no relation to reality.

The first issue is simply volume. Introducing names and socioeconomic data on clients is time-consuming. The information may be computerized, but there are usually incompatibilities between the old and new MIS in the type of information required or the format in which it is stored. While it is often tempting to transfer incomplete data electronically and then enter the missing data manually, this approach requires more attention from a technician and can be costlier than simply assigning lower-cost data entry people to enter all the data manually.

Financial data are an even bigger problem. The actual data in the institutions is often flawed, sometimes seriously. Installing a new MIS then becomes an exhaustive auditing exercise — not necessarily a bad thing, but it adds substantially to the cost of the MIS. Initial balances in the general ledger need to be matched with the detailed sub-ledger balances for all accounts. Financial data should be entered in small batches of fewer than 50 accounts. The totals for the batches need to be checked manually against hard copies from the old system and compared with computer-generated listings from the new system.

A third common issue with loan portfolio modules, and the most serious source of problems during data transfer, is incompatibility in loan treatment between the old and new systems. The new MIS needs to treat a loan midway through repayment predictably — an institution can't simply change its policies midway through a contractual arrangement. But the incompatibility is sometimes irresolvable. Institutions with fairly rapid loan turnover (say, less than six months) might be best off using the old system to track outstanding loans until they are repaid and entering only newly approved loans in the new portfolio package.

10.7. Training

A full-featured MIS is complex, and its implementation can require big shifts in an institution's operating procedures. Installation of any new system must be coordinated with extensive training for all staff. Training can take anywhere from one or two days to one or two weeks, depending on the system's complexity and the number of staff to be trained.

Users should be divided into training groups, usually by department. The training for each group should focus on issues most relevant to its area of operation, but all users should receive a good overview of the system's general operations. The duration of training varies, depending again on the system's complexity and on the staff's experience with similar systems. It is best to break training up into daily sessions of one to two hours.

The training curriculum should include the following:

- System set-up, maintenance and back-up;
- Opening and closing client accounts, and altering and correcting client information;
- Entering information for each type of service or product;
- Following clients through application processes;
- Correcting transactions recorded in error;
- Daily, monthly and annual closings;

- Statement and report generation;
- Use of report writers;
- Security and internal control procedures;
- System restart and data recovery procedures.

Two last points about training. Training, obviously, is important. However, training is often short-changed. Talk to any trainer who is charged with introducing a new application in a company and you will typically hear about how implementation schedules have slipped, the testing environment the trainer was promised in order to develop training materials remained in flux right up to the day of training, and more. The importance of training in such an environment is often minimized. As a project leader or executive director of an agency, it will probably fall in your lap to instill a greater level of importance in training. One way to think about training in this situation is like a first date. After much anticipation, your end users are finally "meeting" the full application. How well or badly that first training experience goes will leave a lasting impression on the end-users, far beyond the end of training. First impressions are lasting impressions.

The second point about training is to be open to different ways of providing ongoing training, either formally or informally. User groups, either face-to-face or through a web page can be very helpful as an ongoing resource for end-users to solve their own problems as well as a contact point to reach technical support staff. Such groups often need some help from leaders in the organization — a helping hand to get started and an occasional check-in to see if the group is still providing some value.

10.8. Implementation: plunge, pilot, phased, or parallel?

There are four possible implementation routes: plunge, pilot, phase, or parallel.

Plunge means that on a specified date the old system is dropped and the new system is used exclusively. This approach depends on having done extensive testing beforehand. An advantage of the plunge strategy is that users must adapt; they have no choice of continuing to use the old system. However, the approach is considered quite risky, as any errors at all mean that staff members have no means of performing their work, and there is no means to compare new reports with old reports to validate accuracy of the new system.

The **Parallel** approach refers to running the old system and the new system concurrently for a period of time, allowing the results to be examined. Advantages to this approach include the ability to compare new and old data in order to catch errors in programming, and the improved sense of security that users feel when not forced to make an abrupt change. However, disadvantages to this approach are substantial, including the significant extra effort demanded of staff as they input data twice. In addition, it is often a challenge to compare reports from the two systems as the systems will not be fully compatible and many differences are not due to errors but are to be expected and even desired. (Remember, you have probably built in new ways to do some processes in the new system). Often, extra staff is brought on during this time to handle the extra load — a significant expense.

The **Phased** approach is where the functionality of the new application is "phased in" — for example, the volume of transactions handled by the new system is gradually increased (e.g., new loans are tracked on the new system while old loans are tracked on the old system). Or, if there are 3 major modules in the new application, the modules are brought up one at a time. Advantages include allowing staff time to get used to the new system and the ability to detect and correct errors without as much danger of significant down time. Disadvantages include the prolonged time period during which data is spread between two systems. The Phased approach is often considered unnecessary when the institution has a relatively small amount of data or the complexity of the new application is not considered to be very high.

The **Pilot** strategy is useful when a number of installations of the same system are expected, as in a branch office structure. A conversion is done at one site (using any of the plunge, parallel, or phase options described above). After a testing period, the system is then implemented at other sites. The advantage of this approach is that any problems are limited to a single installation. However, the approach has the limitation that it will take longer for the entire institution to be converted and aggregation of data will be problematic until the full transition is complete.

How to choose the right approach

One should decide which implementation approach to use early on, as part of the design phase. There are several reasons for deciding this early. For example, the whole project schedule is very different if one chooses a phased approach (if different software modules will be introduced over time) instead of a pilot approach (where the entire application will be introduced all at once). Also, one's testing process tends to be especially rigorous if a plunge implementation is selected (the old system is "turned off") as opposed to a parallel implementation (where both the old and the new system are run together for a month or two).

The appropriate implementation method depends on the scale of the institution, the reliability of the new system, the willingness to risk being without a system for a period of time should things go awry, the attitude of staff toward change, and cost. The parallel approach is the most conservative, most dependable, and most expensive. The plunge is the highest risk but least expensive (providing the new system works reasonably well). However, both the phased and pilot approach have advantages as well. Each one offers a way to reduce "exposure" and limit the impact of any problems to either a small number of people (if using the pilot approach) or a small part of the overall application (if using the phased approach). The other advantage of these two approaches is that a small core of technical support staff can handle the load. If, for example, one used the plunge or parallel implementation process and there were many problems, the technical support staff would be quickly overwhelmed.

When considering which option, it may be helpful to think back to the scale of complexity score determined in Section 2.3. For example, an institution scoring an 8 or 9 (out of a maximum of 9) would likely conclude that the "plunge" option is far too risky.

It is important to run the new system in parallel operation with the old one to insure that the new system runs reliably and that its calculations and processes are accurate and compatible with loan contracts. The institution may need to contract additional personnel during this stage or to retain temporary staff that might have been hired for the data transfer.

During the parallel operation, staff should enter as much data as feasible into each system and carefully compare the outputs. Any discrepancies should be evaluated and accounted for. Any errors or bugs in the new system should be carefully documented and corrected.

The parallel operation should generally continue for at least two months, so that nearly every client will have made at least one payment and the system will have gone through two month-end closings. Once the institution is satisfied that the new system is performing well, the old system can be discontinued, but all printouts and data files should be carefully stored for future reference.

10.9. Data back-up policies

Everyone knows it is important to make back-ups, but virtually nobody has a totally secure back-up policy in place. It is important but not urgent, and therefore, implementation gets postponed, sometimes until it is too late. The following case study both emphasizes the importance of a good back-up policy and provides guidance for establishing a solid policy in your institution.

Case Study: The Day ISED's Office Burned Down

Nothing makes an information technology person like me grab his cell phone and dash out of the house quite like finding out on the 6:30 a.m. breaking news that his office complex is currently a three-alarm fire! This happened to our organization recently, and here's a snapshot of what I remember asking myself as I hurried to my car:

"Did I remember to close up the fire safe last night?" "Did I remember to lock it?" "Did I remember to lock the inner media vault of the fire safe?" "Was the back-up tape from the previous night a good one?" "Did the news helicopter footage show the fire on the third floor or was it still contained on the second?"

Since we fortunately managed to get through this disaster and have the benefit of hindsight to reflect on what factors helped us survive it in good shape, let me jump straight to some of our "lessons learned." Some of them are common sense, others we learned the hard way.

Lesson 1: Contact all employees and remain optimistic

For us, the fact that our employees offered to use space in their homes as offices, that they were able to borrow computers from friends, family and colleagues, and that they raided a nearby office supply store for paper, pens, floppy disks, boxes and other essentials — these positive and constructive steps by all of our staff kept us going. People will be resourceful when there is a disaster; just do your best to support them.

- You're going to need to make a lot of phone calls while you watch your office go up in smoke. Make sure you:
  - Have a cell phone with you and the ability to recharge it during the day.
  - Have a hard copy of all emergency phone numbers stored off-site.
  - Finish contacting every employee and identify what they would have needed to do today had the fire not occurred.
  - Locate copies of those documents; if possible, have them e-mailed to your temporary server (see Lesson 3) and transfer them to floppy disks for distribution.

Lesson 2: Re-route incoming e-mail

If your organization uses e-mail as much as ours, you will want to have a plan for re-routing all incoming e-mail during and immediately after the disaster.

- Call your Internet Service Provider and arrange for the Internet circuit to be re-routed to a computer in their office; ask them to help you.

- Have an e-mail system installed to pickup all inbound mail.
- Have your Web pages copied to it by your Web developer.
- You should consider using the same people to recover your Internet service that installed and configured your original server. They will have already recorded your license keys, for example, and may be able to provide temporary equipment as part of your recovery plan.

Lesson 3: Prepare for the worst case scenario

Develop your disaster recovery plans starting with these assumptions:

- You wake up one morning and the office building is just a pile of ashes;
- There may not be any support services on site such as electricity, seating and shelter. We eventually located a temporary shelter at a nearby church.

Lesson 4: Get and use good fire safes

When I got to the office that morning, there were already four fire trucks hard at work, and the entire building was getting ready to collapse. It was obviously going to put our fire safes to the test. There was also going to be a big lake of filthy water covering whatever was left, and I wasn't sure our safes could keep our back-up tapes dry.

- Store current data on back-up tapes in a "media" fire safe designed to protect tapes from any heat;
- Put that safe itself inside a larger fire safe;
- Use premium grade back-up tapes and upgrade them every year.

Lesson 5: Always have offsite storage for back-up data

There's just no reason to go through the anxiety of what we faced as our entire database sat under eight feet of water in a burned-out shell of an office building. Make sure that your disaster recovery plan includes an offsite facility where you keep your back-up data.

- Your offsite storage should contain everything you will need to rebuild your computer software infrastructure.
- Buy a "media-only kit" from your software vendors so you can make a back-up copy — it's cheap and worth the investment.
- All license keys should also be photocopied and stored there.

Lesson 6: Review your insurance for equipment annually

Be prepared for the fact that no matter how much insurance you have, it will never cover everything should the building be turned to cinders.

Lesson 7: Recovery plans should be reviewed to some degree every day

Whenever something new arrives or changes (e.g., you get some new software or a new server account), evaluate how it will be protected and recovered — put it into your disaster recovery plan. Update the listing in your recovery plan at the office and at the offsite location.

- Annually run through a mock drill. Take your offsite kit, unpack it and then walk through all your recovery plan steps. Verify that your offsite kit contains everything you will need to recover all of your data.

10.10. System maintenance and MIS audits

The software firm's responsibility does not end with the installation. It must provide reliable and timely support to the microenterprise assistance institution to insure that if the system does go down, it does not stay down long. Support is generally provided by e-mail, phone, fax and Internet-based long-distance control of your computer system.

The cost of support will depend on the system's stability and reliability, with a relatively new system that has not been thoroughly tested requiring much more support. The cost will normally decline as the institution grows more experienced with the MIS and, thus, more capable of solving problems. The cost of revisions and modifications needed as the institution evolves and changes its procedures usually is not included in charges for support. Firms will charge additional fees for upgrades of the source code and for customized modifications.

Finally, regular reviews of the program — MIS audits — will be needed to insure that the system continues to function properly, reflects the institution's current policies and procedures, and meets its information and management needs. A review is recommended once every three years.

Case Study: Implementing the new system

Once Women's Initiative decided on a custom module, they spent the next 18 months developing the system. Work consisted of data-entry form development (getting the data in), report development (getting the data out), and much refining and cleaning of existing data. The report development phase is still underway. The consultant role was very much to coordinate the technical side of the project: programming and system development as well as managing the ever changing user needs. The biggest challenges have been finalizing the hard copy informational needs of the organization fast enough not to need the consultant to be constantly updating the forms; maintaining project momentum during times that the volunteer consultant was not available; and establishing enough resources (staff and financial) to maintain and continue the project until it is complete.

Overall, Women's Initiative has been very pleased with both the process and the results. Their hard work has resulted in a custom database system that allows for historical data management and a high degree of use by all WI staff.

Chapter 11: Conclusion

Well, you deserve to be congratulated! If you've invested the effort to make it this far, then you must agree that managing information properly is essential to the success of your institution.

Throughout this Manual we have provided information on how to establish a sound management information system for a microenterprise institution. The issues are many; they are complex; and they are highly interrelated. In many cases, you have seen that setting up a good information system may necessitate a significant restructuring of the institution, reworking staff responsibilities (sometimes even some staff qualifications), redesigning work processes and information flows, revising and rationalizing financial policies, making significant investment in computer technology — the list goes on. Our experience has shown over and over that information systems can perform only as well as the institutions they model. As information really is at the core of our work, it shouldn't be surprising that introducing new information systems can affect the organization to the very core.

As we stated at the beginning, if you remember nothing else from this Manual, you should remember the following: MIS is not simply a software program. Your institution's information needs are met not only by software, but also by thinking through and rationalizing how your institution collects, processes, transfers and uses information in operational and strategic decisionmaking.

If you haven't already done so, we encourage you to look through the ***2002 FIELD MIS Software Review.*** The reviews will show you what is currently available on the commercial market. There is no doubt that the software products have improved in recent years and should be expected to continue improving. As practitioners and vendors work together to bring software supply and demand closer together, they should concentrate on the following three factors:

- *Thorough identification of information needs.* Following the guidelines presented in this manual will assist practitioners in clearly identifying their information needs.
- *Effective communication between management and systems people.* Understanding the basic MIS principles and the key concerns of practitioners as presented in this manual will improve communication between program managers and systems developers.
- *Realistic expectations about information technology.* Information users often wonder why they can't have the information they want, when they want it. This manual has attempted to educate information users about what is possible and how much effort it takes to get good information out of a system.

If you have followed the MIS Selection Process as outlined in this manual, you will know that your information needs are a moving target. Needs constantly evolve as an institution matures and as situations change. You will find yourself revisiting the issues addressed in this manual, most likely sooner than you would prefer. In fact, it is advisable to perform a general reassessment every twelve to eighteen months. The good news is that a review takes only a fraction of the time that it takes to work through the process the first time.

In sum, we hope that your institution has benefitted from the use of this Manual. Management in those institutions which have made the decision to invest the time and resources needed to get an effective MIS, and who now have access to reliable and timely information, avow that it was one of the most important decisions they ever made.

Annexes

Annex 1: MicroTest Indicators

Section 4.3.4 described the MicroTest Framework, which was initiated in 1997 by and for microenterprise development practitioners interested in documenting and improving upon the level of performance in the microenterprise field.

The MicroTest Framework focuses on categories of outputs by which microenterprise development providers can assess their performance. While each category is not appropriate for every program, this framework captures the "pillars" of microenterprise performance. It is designed to capture a range of program performance areas (not client- or borrower-level outcomes data) and to mitigate the tendency to evaluate complex program(s) according to one-dimensional criteria. What follows is a listing of the measures and their definitions.

Measure — **Definition**

1. Total Number of Clients

Total number of clients is defined as the number of individuals who received a significant level of service from your microenterprise program in a given fiscal year. A significant service is one that your program believes can be traced to a client's business or personal outcome(s) after that client exits your program. More specifically, MicroTest defines a client as someone who: 1) **had an active, outstanding microloan or other microfinancing product with the program during the fiscal year** and, or 2) **received at least 10 hours of microenterprise-related training and/or technical assistance from the program during the fiscal year.**

2. Number of Women Clients

This is the number of female clients a program serves. This number should be a subset of the total number of clients.

3. Number of Minority Clients

Minority clients are those individuals who identify themselves as: African-American; Latino/Hispanic; Asian/Pacific Islander, Native American, or another racial or ethnic minority. If your program uses another similar definition for minority status, please use that. We are trying to get a general idea of the number of clients who consider themselves a member of a minority group.

4. Number of Low-Income Clients Based on 100 Percent of HHS Poverty Guidelines

This is the number of clients served by your program in a given fiscal year whose household income places them at or below the national poverty line for that particular year as determined by the Department of Health and Human Services. Household income is based on the number of people in the household who share income and expenses. Someone who lives in the same place but pays for his or her own expenses should not be included in the 'household' number.

Generally, this is the strictest definition for poverty; unlike the HUD guidelines, it is not sensitive to regional variations in income. We have included the HHS tables for you to use in determining the number of your clients who meet this low-income definition; the percent of your total clients who meet this definition is calculated for you. Also, if a client entered your program in 1999 and has an outstanding loan in 2001, you would determine that client's low-income status *at intake,* i.e., according to 1999 HHS guidelines.

5. Number of Low-Income Clients Based on 150 Percent of HHS Poverty Guidelines

Like Measure 4, this is the number of clients your program served in a given fiscal year who *at intake* had a household income level at or below 150 percent of the national poverty line as set by HHS for that year. The same tables that provide annual poverty guidelines for 100 percent of poverty, have guidelines for 150 percent of poverty. Please note that if a client meets the definition for 100 percent of poverty, then that client also meets the definition for 150 percent of poverty, and should be counted here. So, in all cases for a particular year, Measure 5 is *higher* than Measure 4, because it includes those clients.

6. Number of Low-Income Clients Based on HUD Guidelines

This is the number of your clients in a given fiscal year whose household income at intake was at or below 80 percent of the median income for households of the same size in the Metropolitan Statistical Area in which your program operates, as determined by the U.S. Department of Housing and Urban Development for that particular year. These guidelines are available on-line at www.huduser.org. Generally, this number *will include* clients from Measures 4 and 5, because 80 percent of median area income is typically higher than 150 percent of the national poverty line, so that clients who meet the HHS definitions of low-income also meet the HUD definition used here, and are included in this measure.

7. Number of Clients Receiving TANF

TANF clients are those individuals who received Temporary Assistance for Needy Families (TANF) cash support at the time of their intake into your program.

8. Number of Clients with Start-up Businesses at Intake

This is the number of clients assisted by the program in the fiscal year that had a business in operation fewer than 12 months at the time of their intake into your program. A business is considered a business when it has made sales or when the client has taken steps to formalize the business. These steps can include obtaining a business license or retail sales tax number, opening a business checking account, or other step to formalize.

9. Number of Clients with Ongoing Businesses at Intake

This is the number of clients assisted by your program in the fiscal year with businesses that were in operation 12 months or more at the time of their intake into your program.

10. Number of Clients Without Businesses at Intake

This is the number of clients assisted by your program in the fiscal year who did not have any type of business in operation at the time of their intake into your program.

11. Number of Clients Receiving Business Development Training and/or Technical Assistance

Enter here the number of clients served by your program in the fiscal year who received a *significant* training or technical assistance service delivered with the intent of helping said client to develop a business (as opposed to a service delivered with the intent of helping a client to obtain microfinancing). *Training* is generally defined as curricula delivered to business owners or potential entrepreneurs in a classroom or group setting. *Technical Assistance* is generally defined as any individualized or one-on-one consulting, counseling, mentoring or facilitation related to business development or personal development of an entrepreneur.

12. Number of Microloans Disbursed

Record here all microloans for business purposes actually disbursed from your loan fund during the fiscal year. Do not include loans made to your clients by banks or other credit providers. Microloans are defined as loans up to $25,000.

13. Dollar Value of Microloans Disbursed

Enter here the sum of the loans recorded above in Measure 12. Do not include loans made to your clients by banks or other credit providers.

14. Clients Linked to Banks or Other Commercial Credit Providers

Record here the number of your clients you prepared to apply for a loan from a bank or other credit provider during the fiscal year *who actually secured a loan.* If you provide the service of linking clients to banks but do not track who actually receives loans, please do not record anything for this measure.

15. Number of Businesses Started After Intake

Number of businesses started after intake is the number of businesses started by clients in the fiscal year after their entry into your program. A business is considered a business when it has made sales or when the client has taken steps to formalize the business.

16. Number of Microloans Outstanding at the End of the Fiscal Year

This is the number of microloans for which principal was outstanding as of the last day of the fiscal year. These loans may have originated during the fiscal year or in a previous year. This number should include any loans that have been restructured, but not those that have been written off.

17. Dollar Value of Microloans Outstanding at the End of the Fiscal Year

For the microloans counted in Measure 16, indicate the total dollar amount of the principal still outstanding as of the last day of the fiscal year.

18. Actual Range of Microloan Sizes for the Fiscal Year

Enter the amounts of the smallest and largest microloans disbursed during the fiscal year.

19. Average Microloan Size for the Fiscal Year

This figure is automatically calculated by the worksheet, using data that you have already entered. It is equal to the dollar value of loans disbursed during the fiscal year, divided by the number of loans disbursed during the fiscal year.

20. Percentage of Portfolio Loaned to Start-ups

This figure represents the dollar amount of microloans lent to start-up businesses divided by the total dollar amount of microloans outstanding at the end of the fiscal year.

21. Restructured Loan Rate

This is defined as the dollar amount of restructured loans outstanding at the end of the fiscal year divided by the total dollar amount of microloans outstanding at the end of the fiscal year.

22. Loan Loss Rate

This figure is the dollar amount declared non-recoverable and written off, net of recoveries, during the fiscal year, divided by the average dollar amount of microloans outstanding for the fiscal year.

23. Percentage of Portfolio at Risk (from 31 to 60 Days Past Due)

This measure represents the dollar amount of principal outstanding on all loans with payments past due 31-60 days, divided by the total dollar amount of microloans outstanding at the end of the fiscal year.

24. Percentage of Portfolio at Risk (from 61 to 90 Days Past Due)

This measure represents the dollar amount of principal outstanding on all loans with payments past due 61-90 days, divided by the total dollar amount of microloans outstanding at the end of the fiscal year.

25. Percentage of Portfolio at Risk (from 91 to 120 days past due)

This measure represents the dollar amount of principal outstanding on all loans with payments past due 91-120 days, divided by the total dollar amount of microloans outstanding at the end of the fiscal year.

26. Percentage of Portfolio at Risk (greater than 120 days past due)

This measure represents the dollar amount of principal outstanding on all loans with payments past due more than 120 days, divided by the total dollar amount of microloans outstanding at the end of the fiscal year.

27. Percentage of Total Portfolio at Risk (greater than 30 days past due)

This measure represents the dollar amount of principal outstanding on all loans with payments past due more than 30 days, divided by the total dollar amount of microloans outstanding at the end of the fiscal year.

28. Training Completion Rate

This is the number of clients who graduated or completed course requirements, divided by the number of clients scheduled to complete them.

29. Business Plan Completion Rate

This is the number of clients who actually completed a business plan as part of a training curriculum, divided by the number of clients who were scheduled or anticipated to do so.

30. Cost Per Client

This measure represents the average cost of serving a client in the fiscal year. It is determined by dividing the total cost of the program (including both training and technical assistance and credit program costs) by the number of clients served during the year.

31. Cost Per Assisted Business

This measure represents the program's average cost to provide assistance to a business in the fiscal year. This cost is usually higher than the cost per client, because many programs typically assist some clients who do not have businesses and some businesses that have more than one owner.

32. Business Start Rate

This ratio shows the rate at which your clients started businesses in the fiscal year. It is the number of businesses started in the fiscal year by clients who had no business at the time of their intake into your program.

33. Cost Per Business Development Training/Technical Assistance Client

This measure represents the program's cost of serving its business development training and technical assistance clients in the fiscal year. It includes only the expenses associated with the training and technical assistance component of the program, and presents them on a per client basis.

34. Cost Per Participant

This measure represents the program's cost to serve all participants (including clients) in the fiscal year.

35. Cost Per Loan

This measure represents the average cost to disburse a loan in the fiscal year. It is calculated by dividing the operating costs of the credit program by the number of loans disbursed during the year plus the number of clients linked to other credit institutions. This enables the cost of the credit program to be spread over all loans that are generated whether or not they come from a program's own lending facility.

36. Operational Cost Rate

This is another measure of the efficiency of an organization's credit program. It reflects the organization's cost to make and manage loans in the fiscal year. Unlike the cost per loan measure, it includes resources expended both in making new loans and in managing outstanding loans made in previous years. The measure is calculated by dividing the operating expenses of the credit program by the average outstanding loan portfolio for the year. The resulting number represents the cost the organization incurs to manage $1 in its loan portfolio. For example, an operational cost rate of 1.10 would mean that it costs the organization $1.10 to manage each dollar in its loan portfolio.

37. Clients Per Direct Service Provider

This is an overall efficiency measure for microenterprise programs. It represents the number of clients managed or served, on average, by each direct service staff member.

38. Operational Self-Sufficiency

This measure represents a lending program's ability to cover the operating costs of its credit program with internally generated income. It is calculated by dividing the financial income derived from the loan fund by the credit program's operating costs. Financial income is interest and fees paid by borrowers and/or interest income generated by investment of unused loan funds.

39. Short-Term Financial Self-Sufficiency

This measure represents a lending program's ability to cover the costs of its credit program with internally generated income. It is calculated by dividing the financial income derived from the loan fund by the sum of the credit program's operating costs and financial expenses.

40. Training Program Cost Recovery

This measure represents a training program's ability to cover the costs of its training and technical assistance program with internally generated income. It is calculated by dividing income generated from training fees and other fees directly related to the training program (workshop fees, income from sales of training materials) by the total expenses of the training program.

41. Program-Related Income as a Percent of Operating Expenses

This measure reflects the microenterprise program's overall ability to cover its total costs with internally generated income derived solely from its services to program clients. The measure essentially combines Measures 39 and 40. It is calculated by dividing the internally generated income from both lending and training and technical assistance services provided to clients by the total costs of operating the microenterprise program.

42. Net Income from Non-Program Service

This measure reflects the fact that microenterprise organizations are increasingly trying to generate income by providing services to individuals and organizations outside of their targeted client base or through special fundraising events. The net income, or profits, from activities such as consulting and training services to other microenterprise organizations, special events or conferences is then used to subsidize the provision of services to program clients who cannot afford to pay for the services they receive. This measure is calculated by dividing the net income (or profits) derived from these activities by the total costs of operating the core microenterprise program.

43. Percent Program Funding

This measure reflects the degree to which program income is generated internally by fees, etc. It is calculated by dividing total program-generated income by total income of the microenterprise program.

44. Percent Private Funding

This measure reflects the degree to which program income or funding is derived from private sector (non-government) sources. It is calculated by dividing total income from private sources by total income of the microenterprise program.

45. Percent Federal Funding

This measure reflects the extent to which program income or funding is derived from federal government sources. It is calculated by dividing total income from federal sources by total income of the microenterprise program.

46. Percent State Funding

This measure reflects the extent to which program income or funding is derived state government sources. It is calculated by dividing total income from state public sector sources by total income of the microenterprise program.

47. Percent Local Funding

This measure reflects the extent to which program income or funding is derived from local government sources. It is calculated by dividing total income from local public sector sources by total revenues (income) of the microenterprise program.

48. Percent Other Funding

This is a catchall measure that identifies the percent of program funds that are derived from sources other than those included in Measures 43-47. It is calculated by dividing total income from other sources by total income of the microenterprise program.

49. Percent Grant Funding

This measure shows you how much of your microenterprise program’s total income is in grants.

50. Percent Contract Income

This measure isolates the percent of your program’s total income that you receive on a contract basis.

Annex 2: Sample Consultant Terms of Reference

When working with an outside consultant to assist your institution with its information system plan, it is critical to have a very clear Terms of Reference to insure that all parties are clear about the expectations of the work. The Terms of Reference will need to be carefully developed for each instance. The following sample is provided to give a sense for the level of detail and types of issues to be addressed.[33]

TERMS OF REFERENCE
for the development of an
INFORMATION SYSTEM PLAN

Prepared by: ________________________

Revised: ___________________________

1. Introduction

The mission of our institution is…….

The objectives of our institution are…….

Our institution has today virtually no coherent approach to information system management. The proper management of information system will provide the management and employees with fast and accurate access to information, thus enabling them to manage and operate more efficiently and effectively.

To improve operations through the effective use of information technology, our institution has embarked upon a strategic technology planning process aimed at aligning the institution's information technology investment with its core business and policy objectives.

2. Objectives

The goal of the project is to develop a strategic information system plan for the institution that will pursue the following objectives:

- Improve the productivity and management by the appropriate and effective use of information technology;
- Insure staff have the Information Technology skills necessary to achieve the mission and objectives of the MFI;
- Promote a high level of user and management involvement throughout the information system development process;
- Insure demonstrable security of the data holdings;
- Be the institution's road map for creating the proper technology environment to support its mission and to improve its performance;

[33] Thanks to Normand Arsenault of Emerging Microfinance International for providing this sample contract, which he developed for use in his *MIS Practical Guide.*

- Achieve a close fit between business functions and the information systems needed to support those functions;
- Offer an objective assessment of the institution's technical environment in terms of staff, equipment and architecture;
- Define a vision for information technology.

The institution, in cooperation with ____________ (Donor), will periodically review the strategic directions in the plan, identify appropriate projects, set priorities, develop high level implementation plans and then develop yearly operational project plans.

3. Scope of Work

The Scope of Work is not considered exhaustive and modifications will be considered. The consultant should justify all revisions or modifications to the Scope of Work.

The scope of work will be in four phases:

- Phase I: Feasibility Assessment;
- Phase II: Needs Assessment;
- Phase III: Organization Assessment and Project Team; and
- Phase IV: Design and Planning.

Phase I — Feasibility Assessment

During this phase it would be necessary to have a close co-operation with everybody in the institution. It would also be important to incorporate the future needs based on existing plans.

Task 1 – Project start-up

- An initial meeting between institution and consultant to clarify work plan and address any concerns;
- Overview of existing information;
- Development of a work plan;
- Preliminary identification of institution staff members and other stakeholders who will provide relevant knowledge and expertise; and
- Preliminary identification of institution staff members to be interviewed.

Task 2 – Set realistic expectations

Develop, articulate and establish the appropriate level of expectation:

- Describe business functions (Operations, Accounting, Credit, Finance, Marketing, Human Resources, and Organization). Business functions are ongoing activities that support the business;
- For each business function, identify components that may be suitable for automation;
- Produce a high-level analysis of components to evaluate the level of complexity involved in automation;
- Involve institution staff in developing a high-level list of realistic expectations for the automation of each of these components;
- Describe primary benefits the institution hopes to gain from the automation initiative. Identify criteria to evaluate how the project will be successful;

- Determine priorities;
- Develop the purpose statement of the project (purpose statement is a summary of the primary benefits).

Task 3 – Intuitively assess the feasibility of the project

- Before going through the rigor of design and project planning for each of the components, intuitively assess the feasibility of the automation project;
- Review the mission, objectives and culture of the institution and consider any positive momentum to build from;
- Perform S.W.O.T. analysis, which could reveal the institution's Strengths, Weaknesses, Opportunities and Threats;
- Review funding available for information systems development, implementation and management;
- Get input from all stakeholders.

Task 4 – Feasibility assessment results discussions with the institution

The institution will determine how this task will be accomplished. The intent of this task is to insure that the institution has had suitable input into completion of the feasibility assessment. This task may take the form of a round-table with the institution's management and other stakeholders or an open house with the institution's staff members invited at the discretion of the management. Make GO/NO-GO decision.

Task 5 – Draft feasibility assessment report preparation

The report shall document all the findings from Tasks 1 to 4.

The institution's management and other stakeholders will comment on the draft report. Upon receipt of final comments, the consultant should finalize the Feasibility Assessment Report and proceed with the Needs Assessment Phase.

Phase II – Needs assessment

Task 1 – Document the current and the desired situations

Refine the expectations and scope of work further by developing problem/opportunity statements associated with each component and developing the automation requirements:

- Identify the users of the information system and then interview them to understand their current and future business requirements; there are five groups of users served:
  - External users (Factual Information and control),
  - Board of directors (Strategic information),
  - Management (Decision-taking information),
  - Knowledge workers (Tactical Information),
  - Operational workers (Transactional Information);
- Identify Key Contact Persons for each groups of users;
- Prepare for the interviews. Develop a form so that the information can be captured and documented accurately and that good use is made of users' time. Interviews can be formal or informal;

- Examine and analyse the current level of automation versus the automation opportunity for each component in each business function;
- Document the current situation and what is required (the desired situation). Describe inputs, processes and outputs for the current situation and for the desired situation;
- Describe problems and opportunities associated with each component subject to automation. A problem or opportunity statement describes what can be fixed or improved via automation;
- Identify specific requirements and decisions concerning the desirable level of automation. Automation requirements address the problems and opportunities identified in the current situation assessment;
- Develop the documentation by taking inventory and canvassing the staff for information;
- Validate the results of the interviews with the persons who gave the information;
- Meet management for a reality check. Separate desires or "nice-to-have's" from legitimate needs. Match needs with the organization's business plan. Insure that the needs identified are translated into requirements that will help the MFI achieve its business objectives.

Task 2 – Needs assessment results discussions with the institution

The institution will determine how this task will be accomplished. The intent of this task is to insure that the institution has had suitable input into completion of the needs assessment. This task may take the form of a round-table with institution management and other stakeholders or an open-house with institution staff members invited at the discretion of management. Make GO/NO-GO decision.

Task 3 – Draft needs assessment report preparation

The report shall document all the findings from Tasks 1 and 2.

Management and other stakeholders will comment on the draft report. Upon receipt of final comments, the consultant should finalize the Needs Assessment Report and proceed with Organization Assessment and Project Team Phase.

Phase III — Organization assessment and project team

Task 1 – Review organization

Study how automation may impact the functions and structure of the institution:

- Review the organizational structure. Compare MFI to industry standards and norms;
- Identify organizational issues and challenges;
- Identify changes that are needed by the initiative;
- Produce a plan for reorganization and process implementation.

Task 2 – Establish project team

- Produce a plan to establish an automation project team in the context of the organization;
- Identify Project Leader;
- Identify Information and Processes Manager;
- Identify System Manager.

Task 3 – Engage the user community and the technology community

- In collaboration with the Information and Processes Manager, produce a plan to engage the User Community (Communication plan);
- In collaboration with the System Manager, produce a plan to engage the Technology Community (Communication plan).

Task 4 – Organization assessment and project team results discussions with MFI

The institution will determine how this task will be accomplished. The intent of this task is to insure that the institution has had suitable input into completion of the organization assessment and establishing a project team. This task may take the form of a round-table with management and other stakeholders or an open-house with MFI staff members invited at the discretion of the management.

Task 5 – Draft organization assessment and project team report preparation

The report shall document all the findings from Tasks 1 to 4.

Management and other stakeholders will comment on the draft report. Upon receipt of final comments, the consultant should finalize the Organization Assessment and Project Team Report and proceed with Design and Planning Phase.

Phase IV — Design and planning

Task 1 – Prepare a plan for processes implementation

- Take an inventory of existing processes and refine them to support the new levels of automation delivered by the tools. Match the processes with industry standards;
- Define or refine the functions and processes objectives and responsibilities;
- Document revisions thoroughly to prepare for the training required to adopt the new or revised process in the deployment;
- Assess current and future human resources and training needs;
- Prepare a plan for process implementation and training of personnel.

Task 2 – Prepare a plan for technology and software implementation

- Identify current information technology architecture in the institution including, but not limited to:
  - Computers and Peripherals,
  - Internet/Intranet,
  - Local Area Network (LAN) and Wide Area Network (WAN),
  - Client and Network Platforms,
  - Personal Productivity Software,
  - Groupware Software,
  - Database Management Systems,
  - Hardware and Software Management Systems,
  - Accounting and Payroll Software,
  - Loan Tracking Software,
  - Account and Transaction Processing Software,

    - Financial Analysis Software,
    - Customer Relationship Management Software,
    - Geographic Information System (GIS),
    - Human Resources Information Management Systems,
    - Custom applications,
    - Other specialized software,
    - Sharing and access to common data,
    - Data security and protections, and
    - Ideal Information Technology Standards (e.g., applications, platforms, protocols, etc.);
- Review mission and vision for the information technology in the institution;
- Recommend an information technology architecture;
- Determine application priorities, define future governance processes and organizational structure, and develop information technology service delivery recommendations.

Task 3 – Develop request(s) for proposals (RFP)

- Use the automation requirements to develop request(s) for proposals (RFP);
- Issue the request to external vendors and to internal staff;
- Receive and evaluate the responses and choose the solution that most closely meets the automation requirements.

Task 4 – Create a financial strategy

- Choose lease versus buy;
- Estimate cost of implementation;
- Quantify return on investment;
- Create the financial plan.

Task 5 – Design and planning results discussions with MFI

The institution will determine how this task will be accomplished. The intent of this task is to insure that the institution has had suitable input into completing the organization assessment and establishing the project team. This task may take the form of a round-table with management and other stakeholders or an open-house with staff members invited at the discretion of management.

Task 6 – Draft information system plan report preparation

The report shall document all the findings from Tasks 1 to 5.

Management and other stakeholders will comment on the draft report. Upon receipt of final comments, the consultant should finalize the Information System Plan Report.

4. Reporting and meetings

The consultant shall consider the following meetings necessary:

- Project start-up meeting;
- On-going discussions with the institution and other stakeholders;

- Round-table or open-house documenting concerns and inputs;
- Feasibility Assessment Results Discussions with the institution;
- Needs Assessment Results Discussions with the institution;
- Organization Assessment and Project Team Results Discussions with the institution;
- Design and Planning Results Discussions with the institution; and
- Final Information System Plan Report presentation.

Additional meetings may be requested by the institution.

The consultant shall make the following reports necessary:

- Draft and Final Feasibility Assessment Report;
- Draft and Final Needs Assessment Report;
- Draft and Final Organization Assessment and Project Team Report;
- Draft and Final Information System Plan Report;

Five (5) hard copies of each report are to be submitted. Electronic copies of each report will also be submitted to the institution. The software used to develop the databases of information technology information and network diagrams will be determined based on future discussions with the institution and the consultant.

The Information System Plan Report shall include, as a minimum, the following:

- Executive summary;
- Scope and objective of the work;
- The methodology for completing the scope of work;
- The findings and recommendations of the Feasibility and Needs Assessment;
- The findings and recommendations of the Organization Assessment and Project Team;
- The implementation plan including:
  - A project plan with costs and a schedule including a plan for processes implementation and a plan for technology and software implementation; and
  - A training program with costs for users and information technology support.
- Detailed network diagrams illustrating current and recommended information technology architecture and organizational structures; and
- Appendices including:
  - A list of all references used,
  - A list of interviewees,
  - Sample questionnaires used, and
  - Relevant correspondence.

5. Schedule

A schedule has not been established to date and will be open to negotiation prior to signing of the contract.

Annex 3: Table Structure Exercise

The following hypothetical example allows you to apply the database concepts introduced in Section 2.3.2. Review the marketing materials presented by the software vendor, noting what claims they make for their product. Then study the table structure and the field descriptions and determine what limitations the system has because of the table structure.

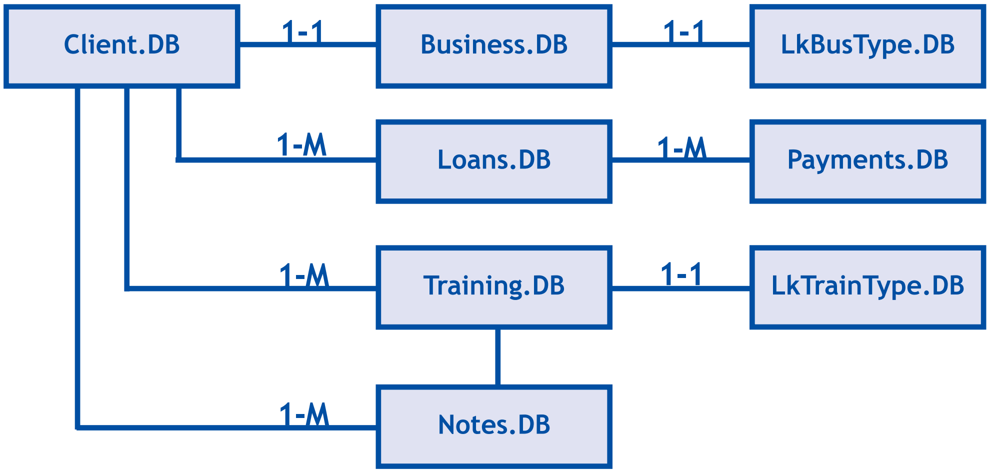
MicroAssist: The integrated MIS solution for your Microenterprise Assistance Agency!

MicroAssist is a state-of-the-art application that tracks everything you want to know about your microenterprise clients. In one integrated package, MicroAssist assembles information about:

- Your clients and their businesses;
- Contact information for all parties, available at your fingertips;
- Socioeconomic/Demographic information to understand who you are working with and to determine eligibility;
- How to keep track of the impact you are having, by monitoring household income, job creation and more;
- Tracking your loan portfolio, including disbursements, income earned and delinquency;
- Tracking your training and technical assistance services by client;
- Providing contact management through an integrated notes module; and
- Much, much more.

Call us today to try out MicroAssist in your agency!

Basic Table Relationships for "MicroAssisst"



Field Descriptions by Table

CLIENT.DB
Social Security #
LASTNAME
FIRSTNAME
MI
Salutation
Address1
Address2
City
State
Zip
County
Tel1
Tel2
Fax
Contact1 Last
Contact1 First
Contact1 Tel
Contact2 Last
Contact2 First
Contact2 Tel
Application Date
Application Status
Granting Source
Birth Date
Age
Race/Ethnicity
Gender
Selective Service
Citizenship
Homeless?
Handicapped?
Monthly Income
Household size
Public Assistance
Assistance Type
Case #
Assist Start Date
Current Assistance Amount
Year-end Amount
Long-term AFDC?
Job Program Participant?
Unemployment Comp?
UC Amount
Veteran?
Education status
Diploma?
GED?
Degree
Employed?
Beginning hourly wage
Post-training hourly wage
Year-end hourly wage
Family Size
Annualized Fam Inc
Econ Disadvantaged?

BUSINESS.DB
Social Security #
EIN
Business ID #
Loan ID #
Business Name
Description
Professional License #
License Type
Incorporation Status
Product/Service Type
Address1
Address2
City
State
Zip
Biz Address1
Biz Address2
Biz City
Biz County
Biz State
Biz Zip
Urban-x
Rural-x
Location Code
Business Telephone
Business FAX
Biz Status
StartUp-x
Existing-x
Business Open Date
Business Close Date
Active?
Biz Type
SoleProp-x
Partnership-x
Corporation-x
Other-x
Biz Other
SIC
Revenue
Number of Employees
PreAssistIncome
Biz Ownership
NonVet-x
VNVet-x
OTVet-x
Black-x
PuertoRican-x
White-X
AmIndian-x
Hispanic-x
AsianPacific-x
MultiGroup-x
EskimoAleut-x
Jobs Created
Jobs Retained
LowIncome-x
YesEntZone-x
NoEntZone-x
YesEmpZone
NoEmpZone

LOAN.DB
Social Security #
LoanType
Loan ID #
LoanAmt
CloseDate
1st Pmt Date
LastPmtDate
LoanRate
Interest Type
Maturity
PmtAmt
PmtFreq
Number of Payments
LoanPurpose
Materials-x
Equipment-x
Supplies-x
WorkingCapital-x
Inventory-x
LoanSecurity
Security Value
YesGuarantor-x
NoGuarantor-x
GName&Add
Prior SBAMicro Bal
Class-x
BizPlan-x
SiteVisit-x
OneOnOne-x
Other-x
FollowTA
Staff Person
LoanStatus
Default Workout
PIF
Charged Off
1st Due Month
1st Due Day
Bill Day
AmtChargedOff
GrantNumber
ProgramLoanNumber
PayeeCode
ReportBy
Userfield1
Userfield2
Userfield3

PAYMENTS.DB
Loan ID#
Payment #
Check Date
Amount Paid
Fund Code
Designation
Begin Bal
Loan Disbursement
Pmt Due
Fee Due
Interest Due
Principal Due
Fee Charged
Interest Charged
TA Charged
Fee Paid
Interest Paid
Principal Paid
TA Paid
Other credit/debit
Ending Bal
Pmt Shortfall
Fee Overdue
Interest Overdue
Principal Overdue
TA Overdue
Total Past Due
Less than 30
30+
60+
90+
120+

TRAINING.DB
Social Security #
DateNum
TAType
TADefinition
StaffCode
Purpose
Hours

NOTES.DB
Social Security #
NoteDate
Note

lkTRAINTYPE.DB
TAType
TADefinition

lkBUSTYPE.DB
BusType
BusDescrip

MIS Cost Estimation Worksheet

Category	Best-Case	Most Likely	Worst-Case
One-Time Expenditures			
Software Costs			
Purchase price			
Customization			
Licensing of other software (operating system, database engine, etc.)			
Configuration			
Data Transfer support (technical assistance, data entry staff)			
Training expenses (trainers, travel, staff overtime, etc.)			
Hardware Expenses			
Servers			
Computers			
Printers			
Back-up power supplies			
Data back-up system			
Infrastucture Improvements			
Wiring			
Workspace modifications			
Other			
Total One-Time Expenditures			
Annual Recurring Costs			
Insurance premiums			
Annual software maintenance			
Annual software licensing			
Annual hardware maintenance contract			
MIS Department staffing			
Annual licensing of other software (operating system, database engine, etc.)			
Total Annual Recurring Costs			

Ratings for Your Institution

Category	Basic (1 point)	Intermediate (2 points)	Advanced (3 points)	Score
Breadth				
Depth				
Scale				
Total Score				

Modules	Rating (Basic, Intermediate, Advanced)	Comments
Loan Portfolio		
Demographics		
Contact Mgmt.		
One-on-One TA		
Training Courses		
Outcomes		
Accounting		

Reporting	Explanation	Our Needs (Ess/Use/NA)				
No. of Loan Reports	Number of basic loan-report formats (not counting filtering criteria as unique reports)					
No. of Other Reports	Number of pre-defined non-loan report formats					
No. of Filter Criteria	Number of filtering criteria provided for reports					
Customizable Reports	Includes a custom report writer					
Can Preview Reports	Allows a screen preview of reports					

Overview Comparison of Software Products

Product	Our Needs				
Modules	Ratings are: 1: Basic; 2: Intermediate; 3: Advanced; N/A: Not Applicable; Opt: Optional				
Loan Portfolio					
Demographics					
Contact Mgmt.					
One-on-One TA					
Training Courses					
Outcomes					
Accounting					
Complexity	Ratings are: 1: Basic; 2: Intermediate; 3: Advanced				
Breadth of Services					
Depth of Information					
Scale of Institution					
Customers					
Software currently used by:					
Decision		To be completed at end of this step			

Ease of Use	Explanation	Our Needs (Ess/Use/NA)				
Keyboard Data-Entry	System is designed to allow rapid keyboard data entry without use of mouse					
Data Validation	Most fields contain data validation to insure accuracy of data input					
On-Line Help	Provides pop-up help or Windows-style help file					
Printed User Manual	Software printed operations manual					
User Tutorial	Documentation includes step-by-step user tutorial					
Training Courses	Vendor provides training					

Security	Explanation	Our Needs (Ess/Use/NA)				
Password Logon	Requires entry of a password to open the system					
User Logon	Requires each user to have a user ID					
Force Password Change	Requires users to change passwords at a specified frequency					
Audit Trail	Tracks changes made to records					
Data Files Protected	Data files cannot be opened from outside of the system					
No. of User Security Levels	Number of different security levels for users					
User-Customized Security	Each user's security access can be customized					
Remote User Authentication	For dial-in users, or for Web-based software, thorough authentication is important					

Integration	Explanation	Our Needs (Ess/Use/NA)				
Link to Accounting	Portfolio data can be linked to other accounting modules					
Export Tables	Allows tables to be exported in a common exchange format					

Technical Specs	Explanation	Our Needs (Ess/Use/NA)				
Web-based Application	Data is stored on the vendor's servers					
Workstation RAM	Minimum RAM requirement for workstations					
Software Language	Language used for development of the software application					
Database	Format used for storage of the data files					
Workstation Operating System	Required for desktop computers of users, e.g. Win 95, NT, etc					
Server Operating System	Required for the server; can differ from the OS on the desktop computers					
Networks Supported	Which networks are compatible with the application, e.g., Windows NT, Novell, etc					

Company Information	Explanation	Our Needs (Ess/Use/NA)				
Year Company Established						
Number of Employees						
Number of Clients	Number of clients currently using the module(s) reviewed in this report					
Demo on CD-ROM	Vendor provides fully functional demo on CD-ROM					
Demo Download from Web	Vendor allows fully-functional demo to be downloaded					
Demo by Remote Access	Vendor provides demonstrations of software by use of remote access software					
Demo Slide-show	Vendor provides a slide-show that gives an overview of the software					
Corporate "transparency"	Vendor gives clear answers to all your inquiries					
Client References	Vendor provides an exhaustive list of clients using his/her products					

Demographic Module		Our Needs (Ess/Use/NA)				
Who is the Client?	Is the focus point of the software the individual(s) who owns the business or the business itself?					
Can Include Non-clients	Allows input of information for individuals/ companies not expected to become clients, e.g., referrals, volunteers, donors, etc					
Allow Identification of Relationships	Individual entries in Clients.DB can be linked to other entries in a variety of different relationships					
Can Include Pre-clients	Allows input of information on prospective clients					
Multiple Clients Per Business	Allows more than one business record to be linked to a client record					

Demographic Module		Our Needs (Ess/Use/NA)				
Linking of Activities	Area activities (loans, training) linked to individuals or businesses?					
No. of Pre-defined Contact Fields	Number of fields dedicated to contact information (address, phone, etc.) per client, not including user-defined fields					
No. of Pre-defined Demographic Fields	Number of fields dedicated to demographic information on the client (not the business)					
No. of User-defined Fields	Number of user-defined fields dedicated to client or business information					
No. of User-customizable Drop-downs	Number of drop-down lists for demographic data which may be user-customized					

Lending Methodologies	Explanation	Our Needs (Ess/Use/NA)				
Individual Lending (e.g, 1 or more loans to 1 client)	Most typical and straightforward methodology.					
Partnership Lending (e.g., 1 loan to 2 clients)	A loan record can be linked to multiple records in the client table.					
Peer Lending with Individual Loans (e.g., 4 loans to 4 clients)	Multiple records in the loan table can be linked to multiple records in the client table, even though there is a primary link of one client to one loan.					
Peer Lending with a Single Group Loan (e.g., 1 loan to 4 clients)	A single record in the loan table can be linked to multiple records in the client table. Client-level detail on the loan is not maintained.					
Participation Loans	The business receives a loan funded from a group of lending institutions. The system tracks the overall loan and the funders' portions.					

Loan Pricing		Our Needs (Ess/Use/NA)				
Fixed Rate Interest	Interest rate does not vary during the term of the loan					
Floating Rate Interest	Interest rate is linked to an external measure and can change throughout the term of the loan					
Compounded Interest	Interest is charged on outstanding interest owed					
Collection of Interest	Interest is collected with each repayment, or interest is collected at disbursement					
360-day Payment Year	Calculations are based on a 30-day month					
365-day Payment Year	Calculations are based on actual number of days					
Allow Specification of Interest Rate	Interest rates are specified for each loan, rather than fixed at same level for all loans disbursed for that product					
Interest Calculations Adjusted for Delinquent and Pre-paid Loans	Does the system allow for different interest charges if the client pays early or late?					

Loan Pricing		Our Needs (Ess/Use/NA)				
Allows Accrued Interest	Interest can be accrued and reported as income before it is paid					
Fixed Up-front Commission	Allows automatic calculation of a fixed amount commission payable prior to or upon loan disbursement					
Percent Up-front Commission	Allows automatic calculation of a percentage commission payable prior to or upon loan disbursement					
On-going Commission	Allows automatic calculation of commissions on an on-going basis, e.g., monthly					
Allows Over-ride of Interest Charges	User may forgive or reduce interest charges					
Insurance Charges	Allows automatic periodic calculation of insurance charges					

Funds Management		Our Needs (Ess/Use/NA)				
Multiple Option for Form of Loan Disbursement	Can loans be disbursed as cash, check, in-kind, or some combination?					
Payment by Cash	Loan payments can be recorded as cash transactions					
Payment by check	Loan payments can be recorded as checks					
Payment by Post-dated Check	Allows input of a series of post-dated checks which can then be transferred to the loan payment register					
Payment by Electronic Transfer	Loan payments can be recorded as electronic transfers					
Payments Entered by Batch	Loan payments are batch processed, with verification of the sum of the transaction					
Assign Loan to Single Fund	Allows fund allocation; a loan can be assigned to a single fund					

Funds Management		Our Needs (Ess/Use/NA)				
Assign Loan to Multiple Funds	Allows fund allocation; a loan can be assigned to multiple funds					
Disbursements in Multiple Tranches	Allows the approved loan amount to be disbursed in multiple installments; ideal for a "line of credit" product					

Repayment Schedule Options		Our Needs (Ess/Use/NA)				
Term Loan with Constant Payment	Can create schedule with constant payments where each payment is combination of principal and interest					
Term Loan with Constant Principal	Can create schedule with a fixed principal payment and a declining interest payment					
Irregular Payments (set by user)	Allows full customization of the loan repayments					
Grace Period, Principal	Allows programming of a grace period on principal repayment					
Grace Period, Interest	Allows programming of a grace period on interest repayment (but no interest calculation)					
Line of Credit	Allows a “line of credit” product, which does not have a repayment schedule					
Single Payment	Allows a single loan payment at the end of the loan term					

Repayment Schedule Options		Our Needs (Ess/Use/NA)				
Balloon Payment	Allows set periodic loan payments with a large balloon payment at the end of the term					
Daily Payment Frequency	Allows programming of daily payments					
Weekly Payment Frequency	Allows programming of weekly payments					
Bi-weekly Payment Frequency	Allows programming of payments every two weeks					
Semi-monthly Payment	Allows programming of payments twice a month					
Monthly Payments	Allows programming of monthly payments					
Bi-monthly Payments	Allows programming of payments every other month					
Quarterly Payments	Allows programming of quarterly payments					
Semi-annual Payments	Allows programming of semi-annual payments					
Annual Payments	Allows programming of annual payments					

Late Payment Options		Our Needs (Ess/Use/NA)				
Penalty on Late Payment	Allows for automatic calculation of late payment penalties					
Optional Over-ride of Penalty	Allows user to over-ride late payment penalties					
Permit Refinancing	Allows user to refinance loans					
Document Securities	Allows input of detailed information on securities offered as collateral					
Loan Write-off	The system allows procedures for writing off unrecoverable loans					

One-on-One TA		Our Needs (Ess/Use/NA)				
Track Total Hours	Allows tracking of total hours of one-on-one TA received by client					
Track Individual Contacts	Allows individual record for each TA event					
Track Duration Per Contact	Records duration of each TA event					
Track Staff Person	Records staff person involved in TA event					
Identify Purpose	Records the purpose of the TA event					
User-Customizable Purpose List Comments Per Contact	Allows user to customize the purpose list. Can record comments for each TA event					
Confidentiality	Allows comments to be marked as confidential					
Plan Future TA	Allows planning for future TA, including date and planned purpose					
Resource Center Tracking	Allows tracking the client's independent use of the institution's resource center; time not assigned to any staff person					

Class-based Training		Our Needs (Ess/Use/NA)				
Input by Class	Allows class attendance to be recorded from a class-orientation rather than a client-information					
Track Class Hours	Allows tracking of number of classroom hours for each client					
Track Attendance/ Absence	Allows tracking of attendances and absences of each client					
Track Facilitator Scheduling	Allows identification of staff/volunteers involved in teaching class, by session					

Contact Management		Our Needs (Ess/Use/NA)				
User-customizable List	Allows user to customize the list of types of contacts					
Assign Staff Person	Records staff person involved in the contact					
Memo Field	Can record comments for each contact					
More than One Memo Per Day	Permits more than one memo to be added with a single date					
Size Limit of Memo	Maximum number of characters per memo					
Confidentiality	Allows memos to be marked as confidential					
Can Edit Earlier Memos	Memos can be edited at a later date					
Can Delete Earlier Memos	Memos can be deleted at a later date					

Outcomes Tracking		Our Needs (Ess/Use/NA)				
Household Income Level	Allows input of household income					
Historical	Data can be tracked historically					
Job Creation	Allows input of job creation data					
Historical	Data can be tracked historically					
Business Financials	Allows input of basic business financial data					
Historical	Data can be tracked historically					
Client Perceptions	Allows input of client perception surveys					
Historical	Data can be tracked historically					

Overview Comparison of Software Products

Product->	Our Needs				
	Modules				
Loan Portfolio					
Demographics					
Contact Mgmt.					
One-on-One TA					
Training Courses					
Outcomes					
Accounting					
	Complexity				
Breadth of Services					
Depth of Information					
Scale of Institution					
	Customers				
Software Currently Used by:					
Decision					

Annex 5: Advanced Hardware Tips

Disclaimer: This document discusses strategies and brief overviews of various topics. It does not necessarily convey all options available or combinations of existing technology. Any reference to existing vendors does not convey a recommendation for or against that particular vendor. Finally, this information was compiled during Fall 2001. Given the rapid changes in computer technology and pricing, be sure to check any information provided below against current technology standards and current pricing.[34]

On the overall system

Settle on one vendor. Try to stick with one vendor for all your computer purchases as much as possible. One vendor sourcing can ease a small organization's maintenance procedures. You won't have to remember who to call for computer troubles when different systems encounter troubles. This should allow you to reduce overall information technology costs and maintenance.

Compare systems even within the same vendor. If you are careful and keep an eye out, you can, for the most part, get a higher-end system for a little more cost than a mainstream system. Most vendors such as Dell and Gateway add components to their higher-end systems, such as DVD's, high-end video cards and CD-RW's that you don't necessarily need in a typical workstation.

What size "box"? Those micro-towers and micro-desktops may look cute but expansion to that system will be limited. If you later decide to add a CD burner (CD-RW), you may have to remove the existing CD-ROM in order to do that (check how many "bays" the unit contains). Also, the motherboard may only have two or three expansion slots, so adding internal devices such as a SCSI controller, network card or modem may be limited. Also your upgrade avenues such as motherboard and/or processor replacements may be more restrictive.

Should I get the extended warranty? Unless you have an information technology person who is comfortable in working with hardware, buy that three-year **on-site** warranty. There are several types of extended warranties available ranging from five-hour to three-day response times to on-site or carry-in service. Usually you can find a three-year on-site extended warranty from major vendors for as little as $99. What this type of warranty means to you, for the most part, is that for the next three years if any hardware malfunctions on that system, the vendor will send someone to your office the next business day with the replacement part and install it at no charge to you. Talk with your vendor about the price differences among the warranties and what they could mean for you.

Buy now or buy later? Look for the best deal that you can find at the time you need the new computer, make your decision and move on. The volatility of the PC market is such that in four weeks a different computer system may be available for less, but when you need that computer today, that will not help you. With the right shopping you are going to purchase a system that can be upgraded later anyway, if need be.

[34] This annex was written by Todd Schuldt and Tim Shea, members of the FIELD MIS Advisory Task Force.

Buy for the long term. Unless you are a for-profit company, you are most likely going to be using that computer workstation until the day it literally dies. By spending just a bit more for an upgradeable computer, you'll be able to keep that computer alive for much longer than if you try to save a little bit during the initial purchase. Computer advances are leveling off; if you buy a good unit now, rather than a unit that is already reaching the end of its useful life, you should be satisfied with your purchase for years to come.

What about leasing? Alternatively, consider leasing your computer equipment. There are two advantages. Firstly, there is less up-front cost. Secondly, it avoids the organizational frenzy every three or four years when you need to justify a big purchase of new computers. Lease payments become a standard operating expense not very different from a monthly phone bill. Keep in mind though that you will need to overlap lease agreements slightly to minimize the disruptions of changing over almost all of your systems at once at lease end. If possible stagger your leasing agreements so that no more than 50 percent of your system's leases are up at the same time.

On specific components

PIII or PIV? Today, most entry level Pentium IV systems truly cost as little as 10 percent more than their high-end Pentium III counterparts, as long as their other components are similar. Even on their worst day, an entry-level Pentium IV system can outperform a high-end Pentium III. Both AMD's Duron and Intel's Celeron/Pentium III are pretty much at the end of their life cycles and will not ever enter the performance range of the Pentium IV systems.

Spending that extra 10 percent on a Pentium IV system will mean that after a few years the high-end Pentium IV processor will be available at a fraction of the current cost. You can at that time replace just the processor within the computer and boost its power anywhere from 25 to 75 percent. Example: One year ago a Pentium IV processor running at 1300 MHz cost around $425, but can be purchased today for as little as $130. A word of caution though, if you purchase a Pentium III computer you will not be able to upgrade it to a Pentium IV without also changing the motherboard (and possibly the case itself) as well. It can still be cost effective to do this though, as motherboard replacements are fairly easy and inexpensive. Using current pricing: Assuming we start with a decent Intel Celeron 500 MHz-based computer and spend $130 for a Pentium IV 1300 MHz processor, $155 for a Pentium IV motherboard capable of using SDRAM, $45 for 512 MB of PC133 SDRAM memory, the cost still equals a lot less than a new computer tower and purchases almost triple the power of the existing computer.

Integrated Video versus external video. Integrated video means that the video card is actually part of the motherboard. Integrated video was developed and became popular when memory costs were very high and memory was scarce. It shares the main system memory on the motherboard for these tasks. It is slower than dedicated video memory chips and requires several other systems of the motherboard in order to function. All this leads to a cheaper initial cost, but at the expense of system performance. There have been reports that integrated video systems can rob as much as 25 percent of the overall system performance possible in some configurations. Dedicated video cards on the other hand are self-contained. They have dedicated memory chips that are generally much faster than the memory on the motherboard and can process the video information at a higher rate, thereby allowing your CPU to concentrate on other tasks. Now there are exceptions to this rule, most notably in laptop computers. Laptops, however, have the higher speed dedicated memory on the motherboard in addition to the regular system memory. The easiest way to determine if your integrated video card is using shared memory or dedicated memory is to look at the system control panel on your windows machine and find the total memory reported. If it is one MB less than your installed memory then you are utilizing shared memory.

Integrated audio versus sound card. There is nothing wrong with the less expensive integrated audio systems on newer motherboards. You don't need that latest audio card with Dolby Digital 5.1 surround sound in the office with powered sub-woofers. Most integrated audio is capable of sounding better than car audio anyway.

SDRAM versus DDR-SDRAM versus RDRAM. Acronyms change fast with memory chips, and it is easy to be intimidated. Think of the memory types as trucks on a highway:

- **SDRAM:** SDRAM memory is 64 lanes of traffic with semi trucks running side by side at 133 mph to deliver your data to your computer system.
- **DDR-SDRAM:** This stands for "Double Data Rate SDRAM." The same trucks are running up and down the same highway, but they have two trailers behind them instead of one. Therefore they can get twice as much work done in the same amount of time. (Your computer must be equipped with a special motherboard chipset to do this.)
- **RDRAM (also called Rambus):** This is a completely different kind of road, with only 16 lanes of traffic rather than 64. However, each truck is traveling at 800mph! The end result is that this road is capable of delivering almost twice as much data as SDRAM. (Your computer must be equipped with a special motherboard chipset to do this.)

Nothing can change the cost of a computer as much as memory can. Currently SDRAM memory that one year ago would have cost you $128 now costs $22. DDR-SDRAM is generally used by the newer AMD Athlon computer systems and is slightly more expensive than SDRAM. RDRAM is specialized memory designed for Pentium IV systems and costs up to five times more than SDRAM. Newer motherboards for Pentium IV systems now allow the much less expensive SDRAM to be used with at most a 10 percent overall performance loss.

As to the amount of memory to purchase, more is better; 256 MB is a good point to start at. A high quality 256 MB memory chip today will cost only $29, but can increase your systems performance by as much as 25 percent.

Network Interface Cards (NIC): IBM versus 3Com versus Generic. High-end intelligent network interface cards, such as 3Com's 3C905 series and IBM's Pro series, off-load the majority of the network tasks from the CPU, allowing them to even further concentrate on other processing tasks. These higher-end cards cost nearly twice as much as their simple NIC counterparts offered from vendors like Linksys and Netgear, but offer better high-network volume transfer rates and lower CPU loads. The actual cost should be around $50 for the intelligent NICs and around $22 for the simple NICs. Consider the alternatives to be two dragsters, where the CPU is the driver and the NIC is the transmission. While they both have the same engine, one is equipped with an automatic transmission (the intelligent NIC) and can do a 200-mph quarter mile with a lot less thinking and effort than the other with a manual transmission (the simple NIC) that can only go 180 mph in that quarter mile. However, 180 may be all you need for your computer tasks, and you may be better off investing that $25 savings into more memory.

AMD solutions. There is nothing wrong with purchasing the AMD processor versions of the modern day PC (Athlon and Duron). In fact, the high-end Athlon designs are pretty much the fastest chips on the market.

Annex 6: Sample Software Vendor Contracts

Contracts are generally provided by the vendor. You will need to read through them very carefully, as in many cases the standard contract will tend to protect the interests of the software vendor more than those of your institution. Be sure that any specific agreements about features and changes have been properly incorporated. Don't be afraid to ask for any modifications to the contract, and don't sign any contract that contains clauses you are uncomfortable with.

DEVELOPMENT AGREEMENT

This Agreement is by and between OurSoftware Computer Solutions ("OurSoftware" or 'us' or 'we') having its principal place of business at 123 Main Street, Ourtown, N.Y., and MicroAssist ("CLIENT" or "you") having your principal place of business at 300 S. Queen, Anyplace, N.Y.

Background:

WHEREAS OurSoftware develops, markets and distributes various software products and provides associated development and support services,

AND WHEREAS Client has licensed certain OurSoftware software products pursuant to a Software License Agreement and has requested that OurSoftware provide certain services for such software,

AND WHEREAS OurSoftware has agreed to provide such services upon the terms and conditions contained in this Agreement,

NOW THEREFORE, in consideration of the mutual premises, covenants and obligations herein, the parties agree as follows:

1. TERM

This Agreement is effective as of the earlier of the commencement of Services, or the date of execution by both parties, and will continue, subject to Section 11, until the date of completion of the Services and payment in full has been received by us. This Agreement shall only be renewed or extended, as the case may be, by mutual written agreement of the parties.

2. INTERPRETATION

2.01 Definitions. In addition to any other terms defined within this Agreement, the following terms will have the following meanings:

"Agreement" means this Agreement and any Schedule executed in connection herewith.

"Business Days" means Monday to Friday, 8:30 a.m. to 5:00 p.m. EST, excluding those Canadian provincial or federal holidays recognized by OurSoftware.

"Modifications" means modifications to or enhancements of the Software (whether in source or object code form) that are requested by you, whether or not such Modifications are included within our regular Software Update or Upgrade path, and for which we charge a fee.

"Schedule" means a schedule appended to this Agreement that describes the particular development services to be provided by us.

“Services” means the development services to be provided by us to you as described in a Schedule. The provision of Services may also include the provision and delivery of Software. In such an event, a reference to Services will include, by reference, the Software.

“Software” means the software computer program as described in the applicable Schedule and all error corrections, Modifications, Updates and Upgrades thereto in machine-readable form, provided to you.

“Software License Agreement” means the software license agreement executed by you and OurSoftware.

2.02 Schedules. The following Schedules shall be appended to and form part of this Agreement:

Schedule A: OurSoftware Services and Client Responsibilities; Fees

3. OURSOFTWARE RESPONSIBILITIES

3.01 Provision of Services. Subject to Section 11, we will provide the Services on a non-exclusive basis and in a professional and workmanlike manner and you agree to accept such Services.

3.02 Appointment of a Representative. We will appoint a designated Services manager responsible for the administration of the Services.

3.03 No Support. Any support provided to you by us for the Software will be in accordance with the Software License Agreement.

4. YOUR RESPONSIBILITIES

4.01 Appointment of a Representative. You are responsible for appointing a designated contact responsible for the administration of the Services. Any change to either the designated contact or contact methods must be in writing and provided to our designated Services manager with not less than two business days notice.

4.02 Your Responsibilities. To insure we are able to provide the Services, you agree that you will perform the responsibilities set out in Schedule A in a diligent and professional manner.

5. CHANGES, DELIVERY AND ACCEPTANCE

5.01 Changing or Adding Services. In the event you wish to request a change in the Services, or if you request that we provide professional services outside the scope of the Services specified in this Agreement, you agree that you will prepare a written change request. We will evaluate and respond to any change request as quickly as our commitments to you and others permit and will advise you in writing of any impact on the cost and delivery of Services as a result of any proposed change. We each agree to negotiate in good faith any changes to costs or delivery of the Services arising from a requested change. Upon written confirmation from your representative, we will proceed with the change, at the price and terms agreed upon and the applicable Schedule will be amended in writing accordingly.

5.02 Delays. If the timetable for performance of any Services is delayed as a result of a delay by you in the performance of your responsibilities, an event of *force majeure* or a change in the scope of Services to be provided, then the timetable for performance of any Services will be extended for the period of time that the Services delivery is delayed as a result of such events and we will not be considered to be in breach of this Agreement.

5.03 Delivery. We will use reasonable efforts to deliver all Services as set forth in each Schedule. However, you acknowledge that we obtain certain services from third party providers and, as such, are not responsible for and cannot control their performance or Service delays caused by them. In the event of a delay, we will advise you as soon as possible of a new date for performance of the Services.

5.04 Acceptance. You acknowledge and agree that you are responsible for insuring that all Services performed and Modifications or Software received are accurate and meet your requirements. You will be deemed to have fully accepted any Software or Services delivered to you upon the earlier of 30 days from the date of delivery by us or your execution of the acceptance documentation provided to you.

5.05 Marketing Initiatives. Upon reasonable request by us, you agree to act as an account reference for new or prospective customers.

6. PAYMENTS

6.01 Fees and Expenses. In consideration of our performance of the Services, including billable travel time and associated expenses, you agree to pay to us the fees and expenses as calculated and invoiced in accordance with each Schedule. If not indicated in the applicable Schedule, all payments will be due and payable 30 days from the date of our invoice to you.

6.02 Taxes. You agree you are responsible for payment of all duties and foreign, federal, state, county and local income taxes, value-added taxes and other taxes, or amounts in lieu thereof, and interest thereon levied or based on amounts chargeable to or payable by you (excluding taxes on our net income). In the event any payments required to be made by you are subject to applicable withholding tax that you are required to deduct from such payments, you agree to promptly provide us with receipts issued by appropriate government authorities for all such taxes withheld or paid by you and you will fully and promptly cooperate with us (at our expense) to provide such information and records as we may require in connection with any application by us to obtain available tax credits.

6.03 Late Payment. If your payments to us are more than thirty (30) days in arrears (as calculated by the date appearing on our invoice), we will give you written notice that you are responsible for payment of all outstanding amounts and finance charges. If the outstanding amounts are not paid within fifteen (15) days of this notice, we have the right to (i) suspend performance of any Services until all amounts are paid in full or such breach is remedied to our satisfaction; and (ii) terminate this Agreement. You agree you will pay all costs including reasonable attorney's fees, incurred by us in collecting overdue amounts. In addition, if you are in arrears to any extent, we may withhold the provision of further Services until all arrears have been paid.

6.04 Currency. If you are located in Canada, all amounts referred to in this Agreement and all payments to be made by you will be made in the lawful currency of Canada. If you are located outside of Canada, all amounts referred to in this Agreement and all payments to be made by you will be made in the lawful currency of the United States.

7. CONFIDENTIALITY AND PRESS RELEASE

7.01 Confidential Information. Each of us agrees to keep confidential the terms and conditions of this Agreement, and any and all information with respect to the other party that it has received or may in the future receive in connection with this Agreement that is not otherwise available to the general public without restriction, including without limitation, the Software, product information, feature sheets, your PIN and any pricing information provided by us to you, and any of your data submitted to or accessed by us. Confidential Information does not include information that: (i) prior to or after the time of disclosure becomes part of the public domain through no fault of the receiving party; (ii) is disclosed to the receiving party by a third party under no legal obligation to maintain the confidentiality of such information; (iii) is in the possession of the receiving party at the time of disclosure without any obligation of confidentiality; or (iv) is independently developed by the receiving party prior to or independent of the disclosure. A receiving party may disclose Confidential Information pursuant to government or court order or other legal process, provided that the receiving party gives the disclosing party prompt notice of such requirement and takes all reasonable steps to prevent or limit such disclosure.

7.02 Further Treatment of Confidential Information. Each of us agrees to hold all Confidential Information in trust and confidence for the other and not to use same other than as expressly authorized under this Agreement. We each agree not to disclose any such Confidential Information without the prior written consent of the disclosing party to anyone other than the receiving party's employees, agents or contractors who have a need to know same to carry out the rights granted hereunder and with whom the disclosing party has a valid non-disclosure agreement. Each of us agrees to protect the other party's Confidential Information with the same standard of care and procedures which it uses to protect its own trade secrets and Confidential Information of like importance and, in any event, will adopt or maintain procedures reasonably calculated to protect such Confidential Information.

7.03 Equitable Relief. In recognition of the unique and confidential nature of the information disclosed by each of us, it is agreed that each party's remedy for breach by the other party of its obligations under this Section 7 will be inadequate and the non-disclosing party will, in the event of such breach, be entitled to equitable relief, including without limitation, injunctive relief and specific performance, in addition to any other remedies provided hereunder or available at law.

8. OWNERSHIP AND COMPANY LICENSE

8.01 OurSoftware Ownership. We or our licensors own all right, title and interest, including all copyright and all other intellectual property rights whether now known or hereinafter discovered or developed, in and to the Services, Software and Modifications, or any other products or materials provided hereunder or in connection herewith.

8.02 Modification License. You are granted a non-exclusive, non-transferable, limited license to use the Modifications supplied under this Agreement solely in connection with your use of the Software as permitted pursuant to the Software License Agreement.

8.03 Assignment. To the extent you may have obtained any rights in or to the Modifications, you hereby sell, transfer, convey and irrevocably assign in perpetuity, all such right, title and interest in and to any such Modifications, including waiver of all moral rights therein, as of the moment they are created.

9. WARRANTY AND LIMITATION OF LIABILITY

9.01 Warranty. We warrant that the Modifications provided to you hereunder will function in accordance with the specifications as set forth in the applicable Schedule for a period of thirty days from the date of acceptance by you (as defined in section 5.05) provided that all use of the Modifications is for the purposes and in the environment for which they were designed and in accordance with such specifications. Your sole remedy and our sole obligation for breach of this warranty is to use our reasonable commercial efforts to revise the Modifications or specifications (as required) to comply with this warranty. EXCEPT FOR THIS EXPRESS LIMITED WARRANTY, THE SERVICES AND MODIFICATIONS ARE PROVIDED BY US OR OUR LICENSORS OR SUPPLIERS TO YOU "AS IS" AND THERE ARE NO WARRANTIES, REPRESENTATIONS OR CONDITIONS, EXPRESSED OR IMPLIED, WRITTEN OR ORAL, ARISING BY STATUTE, OPERATION OF LAW OR OTHERWISE, REGARDING THEM OR ANY OTHER PRODUCT OR SERVICE PROVIDED HEREUNDER OR IN CONNECTION HEREWITH. WE, OUR LICENSORS AND SUPPLIERS DISCLAIM ANY IMPLIED WARRANTY OR CONDITION OF MERCHANTABLE QUALITY, MERCHANTABILITY, DURABILITY OR FITNESS FOR A PARTICULAR PURPOSE. NO REPRESENTATION OR OTHER AFFIRMATION OF FACT, INCLUDING BUT NOT LIMITED TO STATEMENTS REGARDING PERFORMANCE OF THE SOFTWARE, MODIFICATIONS OR THE SERVICES BY OUR SUPPLIERS OR US THAT IS NOT CONTAINED IN THIS AGREEMENT, WILL BE DEEMED TO BE A WARRANTY. WITHOUT LIMITING THE FOREGOING, WE DO NOT MAKE ANY REPRESENTATIONS OR WARRANTIES WHATSOEVER WITH REGARD TO PRODUCTS OR SERVICES FROM THIRD PARTIES AND ASSUME NO RESPONSIBILITY OR LIABILITY WITH RESPECT TO THE APPROPRIATENESS OF YOUR DATA MANAGEMENT SYSTEM OR THE ACCURACY OF DATA CONTAINED IN SUCH SYSTEM.

9.02 Limitation. The entire liability of us or our suppliers or licensors and your exclusive remedy with respect to the Services or Modifications, or any other services, products or materials supplied by us in connection herewith or hereunder for damages for any cause and regardless of the form or cause of action, whether in contract or in tort, including fundamental breach or negligence, will be limited to your direct damages and will not exceed, in the aggregate, the amounts paid by you to us with respect to the Services giving rise to the damages. IN NO EVENT WILL WE, OUR SUPPLIERS OR LICENSORS BE LIABLE FOR ANY PUNITIVE, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY OR CONSEQUENTIAL DAMAGES SUFFERED BY YOU OR ANY OTHER PERSON, INCLUDING WITHOUT LIMITATION, FAILURE TO REALIZE EXPECTED SAVINGS, ANY LOSS OF REVENUES OR PROFITS, LOSS OF DATA, LOSS OF COMPUTER TIME, OR ANY OTHER COMMERCIAL OR ECONOMIC LOSSES, EVEN IF WE HAVE BEEN ADVISED OF THE POSSIBILITY OR LIKELIHOOD OF SUCH DAMAGES OR SUCH DAMAGES ARE FORESEEABLE.

9.03 No Variation. NO AGREEMENTS VARYING OR EXTENDING THIS WARRANTY OR THE FOREGOING LIMITATIONS WILL BE BINDING UNLESS IN WRITING AND SIGNED BY AN AUTHORIZED SIGNING OFFICER OF OURSOFTWARE.

9.04 Limitation Period. No action, regardless of form, arising out of or in relation to this Agreement or any Modifications or Services may be brought by you more than two years after the completion of the Services to which the action relates.

10. YOUR INDEMNIFICATION

You will indemnify and hold us harmless from all claims, actions, damages, liabilities, costs and expenses brought against, suffered or incurred by us (i) arising out of the use or possession of any Software or Modifications by you or a third party; (ii) our use or copying of any information or material supplied by you; or (iii) our compliance with any of your instructions or specifications with respect to any Services. To assist you in the defense of such claim(s), we agree to provide you with prompt written notice of any claims and to provide reasonable cooperation and assistance (at your sole cost and expense) to defend and settle such claim.

11. TERMINATION

11.01 Termination. This Agreement will terminate in the event of any of the following:

(i) If you are in breach of any obligations under this Agreement, including but not limited to any payment obligations, we have the right, at our sole discretion to: (a) suspend performance of any Services until all amounts are paid in full or such breach is remedied to our satisfaction; and (b) terminate any outstanding Services immediately upon written notice to you, if payment is not made or the breach remedied within thirty (30) days following written notice to you;

(ii) Upon written notice of termination by one party, effective immediately after a receiver has been appointed in respect to the whole or a substantial part of the other's assets or a petition for bankruptcy or liquidation is filed by or against that other or if the other has been dissolved or liquidated or is insolvent; or

(iii) Immediately, in the event your Software License Agreement terminates and you are no longer licensed to use the Software for which the Services are being provided.

12. EFFECT OF TERMINATION

12.01 Effects of Termination. In the event of termination:

(i) We will immediately cease to perform the Services;

(ii) You will perform all payment obligations to us within thirty (30) days of receipt of our final invoice in respect of the Services;

(iii) We will deliver to you all work in progress for which we have been paid in full; and

(iv) Each of us agrees to destroy or deliver up all Confidential Information of the other, except that we may retain any Confidential Information reasonably required for archival, support, back-up, legal or regulatory purposes; provided that all such information will continue to be treated as Confidential Information and such Information will only be retained for as long as its retention is required.

13. MISCELLANEOUS

13.01 Entire Agreement. This Agreement, including any Schedules, constitutes the entire agreement between us concerning the subject matter and supercedes all prior statements, representations, discussions, negotiations and agreements, both oral and written, including your order forms, our acknowledgement of order forms and invoice forms. This Agreement may not be amended or modified except in writing signed by authorized officers of both of us. In the event of a conflict between the terms of this Agreement and a Schedule, the terms of this Agreement shall govern to the extent necessary to resolve such conflict.

13.02 Severance. In the event that any one or more of the provisions of this Agreement is found to be illegal or unenforceable, this Agreement will nevertheless remain in full force and effect, and such term or provision will be deemed severed.

13.03 Independent Contractors. Each of us is an independent contractor. This Agreement does not establish a relationship of principal to agent, master to servant, employer to employee or franchisor to franchisee. Neither of us has the authority to bind the other or incur any obligation on its behalf.

13.04 Non-Hiring. Each party agrees that during the term of this Agreement and for a period of six (6) months thereafter, it will not solicit or hire employees of the other who have performed work relating to this Agreement without the other party's prior written consent.

13.05 No Waiver. A failure by either party to enforce any right under this Agreement shall not at any time constitute a waiver of such right or any other right, and shall not modify the rights or obligations of either party under this Agreement.

13.06 Assignment. This Agreement or any rights granted hereunder will not be assigned, encumbered by security interest or otherwise transferred by you without our prior written consent and any attempt to do so will be void and of no force and effect.

13.07 Notices. Any notice to a party required or permitted hereunder shall be sufficiently given only when provided in writing and either personally delivered to a responsible officer of the addressee or sent via certified or registered mail (return receipt required) or facsimile (with proof of transmission) to the party's address indicated below and shall be deemed to have been received when such notice should have reached the addressee in the ordinary course, provided there is no strike by postal employees in effect or other circumstances delaying mail delivery, in which case notice shall be delivered by facsimile (with proof of transmission).

________________________ ______________________________

OurSoftware Client

13.08 Further Assurances. Each of us agrees to do all such things and to execute such further documents as may be reasonably required to give full effect to this Agreement.

13.09 Force Majeure. Except for obligations of payment arising hereunder, neither party will be liable for delays in its performance hereunder due to causes beyond its reasonable control, including but not limited to, acts of God, acts of public enemy, acts of government or courts of law or equity, civil war, insurrection or riots, fires, floods, explosions, earthquakes or other casualties, strikes or other labor troubles, failure of the Internet or Internet access providers, always provided that the party so relieved of its obligations hereunder provides notice to the other party and takes all reasonable and necessary steps to resume performance of its obligations as soon as possible.

13.10 Governing Law. This Agreement will be governed by and construed in accordance with the laws of the Province of Ontario, excluding that body of law applicable to choice of law and excluding the United Nations Convention for the International Sales of Goods and any legislation implementing such Convention. If either of us employs attorneys to enforce any rights arising out of or relating to this Agreement, the prevailing party will be entitled to recover reasonable attorney's fees. Each of us waives any right, and agrees not to have any dispute under this Agreement tried or otherwise determined by a jury, except where required by law.

13.11 Survival. The provisions of Sections 6 to 10 (excluding 8.02), 12 and 13 will survive the termination of this Agreement until expressly waived in writing by the party for whom they are of benefit or terminated by a further written agreement of the parties.

13.12 Language. The original of this Agreement has been written in English and you waive any right you may have under the laws of your territory to have this Agreement written in any other language. You represent that you have the ability to read and write in English and have read and understand this Agreement. If this Agreement is translated into a language other than English, the English version and interpretation shall govern and prevail. All communications between the parties hereunder shall be in English.

ACCEPTED AND AGREED TO THIS 17TH DAY OF SEPTEMBER, 2001.

OurSoftware	CLIENT
Per:______________________________	Per:______________________________
Name: ____________________________	Name: ____________________________
Title: ____________________________	Title: ____________________________
Date: ____________________________	Date: ____________________________

SCHEDULE A

OURSOFTWARE SERVICES AND CLIENT RESPONSIBILITIES

A. OURSOFTWARE SERVICES:

A.1. We will provide at no additional cost to you:

_ Conversion of existing loan portfolio,

_ Gold-level support,

_ Implementation and training*,

_ Reporting and follow-up,

_ User surveys,

_ Updates,

_ Upgrades.

A. 2. List here any modifications or special features to be enabled.

B. CLIENT RESPONSIBILITIES:

You are required to:

- Provide the appropriate and necessary resources to insure we are able to complete the Services.
- Carry out reviews and respond to requests for approval and information on a timely basis.

- Insure that at least one representative is available during regular business hours to provide such information and assistance as we may require in connection with the delivery of the Services; and insure that we also have available to us personnel familiar with your requirements and with the expertise necessary to permit us to undertake and complete the Services.
- Provide us with timely and accurate information and documentation, as reasonably required by us to perform the Services.
- Obtain and maintain the Software in a proper operating environment, including the proper servicing of such Software and all necessary Updates or Upgrades to such Software.
- Provide a safe area for us to perform any Services to be performed on your site.
- Provide for all power, environmental requirements, supplies, cabling, communications facilities, and all other equipment and facilities required to receive the Services, including high speed Internet access.
- Insure that your data management system and data is accurate and appropriate for your requirements.

C. FEES

Description	Cost

D. PAYMENT SCHEDULE

E. ADDITIONAL WORK RATES

More than one week duration	$xxx/week
More than one day duration	$xxx/day
Less than one day duration	$xxx/hour

Additional copies of this manual can be purchased for $35 by calling the Publications Hotline at 410-820-5338, or by visiting the FIELD Web site: *www.fieldus.org/publications/index.html*

Also available for $20 from the Publications Hotline or FIELD's Web site:

> *2002 FIELD MIS Software Review (forthcoming, October 2002).* This companion document to the MIS manual is an outgrowth of the Findings Report first published by FIELD in 2001. It contains detailed reviews of major commercial software designed for use in the microenterprise field, applying the evaluative approach described in the MIS manual. It is our hope that the software review guide will be updated periodically to reflect changes in the software available for the field.

Both publications can be purchased together for $45.

To learn more about FIELD, please visit our Web site (www.fieldus.org) or contact us at:

FIELD
Microenterprise Fund for Innovation
Effectiveness, Learning and Dissemination
The Aspen Institute
One Dupont Circle, NW, Suite 700
Washington, DC 20036
Phone: (202) 736-1071 • Fax: (202) 467-0790

e-mail: fieldus@aspeninstitute.org

Production Credits:
Design, Graphics
and Composition:
Olmsted Associates
Flint, Mich.

Printing:
The Riegle Press, Inc.
Flint, Mich.

THE ASPEN INSTITUTE

One Dupont Circle, NW
Suite 700
Washington, DC 20036