Training and Supporting Nontechnical People Who Use the SAS System
Kim L. Kolbe, Systems Seminar Consultants

Abstract
Frequently, the people who use the SAS System the most are those individuals who are very accomplished in their respective field, but are far from being full-fledged programmers. This makes the job difficult for the training and user support personnel because often times these users know precisely what they want, but cannot get past the technical details to produce the desired results.

Introduction
Nontechnical/nonprogrammers become frustrated much quicker with the training process than those individuals who have had some programming experience. It’s understandably so, since they have little, if any experience, to draw upon to help them in the learning process. Frustration often results from the user encountering a series of small problems which they are unwilling or unable to deal with. The cumulative effect is a feeling of hopelessness and a decreased level of motivation to continue with the learning process. Minimizing or eliminating as many of these frustrations for the new user will make their experience more enjoyable, and in the end, they will walk away having learned and retained more of the information.

Course Development
An essential ingredient to any successful training course starts with well-written and well-presented training materials. Special attention needs to be paid to course materials developed for those people who have little or no experience with programming languages. For these types of learners it is critical that the materials offer a variety of learning scenarios at varying levels of difficulty, and that they be free of technical jargon.

Learning your first programming language, like learning your first foreign language, is always the hardest. After you learn one, other languages come more easily through a process of association. When learning a new programming language we have to disassociate previously learned words, vocabulary and meanings and learn new definitions and usage as it relates to the programming language being learned.

Developing Materials for Nonprogrammers
The biggest pitfall in developing any type of course materials is that often times materials attempt to meet everyone’s needs and attempt to cover every possible situation and scenario that may be encountered. Individuals learn better when the information is broken into smaller logical modules that are targeted to a specific audience and which cover concrete topics versus something universal and general in nature.

Modularization has been a buzz word in the course development community for the last couple of years. Modular course materials and training are particularly suitable for nonprogrammer-type learners. Modular training breaks a 2 or 3 day training class into 10 or 12 2-hour modules where a specific topic is covered in-depth. Modularization provides some distinct benefits:

* Allows the student to decide what they want to learn and when they want to learn it
* Provides a forum for people to re-learn or re-educate themselves on specific topics
* Student’s tend to retain the information better since it’s applied sooner
* Time away from the job is spread out rather than condensed into 2 or 3 days

When developing course materials also keep in mind that both programmers and nonprogrammers alike will benefit from materials which provide a number of different types of learning activities. Multiple choice and fill-in-the-blank questions intermixed within the lecture materials provide the learner’s a feedback system whereby they can test their knowledge as it is
learned before they progress. Supplemented with short programming exercises at the end of each topic provides a variety and solidifies concepts learned in the lecture. Remember not everyone learns the same way, providing as many different types of learning experiences within the course will allow the student to find the method which works best for themselves. Learning activities with various levels of difficulty will provide student's with the ability to tailor their learning experience.

When developing course materials for nonprogrammers critically examine all the syntax and technical jargon. SETting a data set and the value of a variable have perfectly logical meanings to us as programmers. New users have to re-learn and re-define their vocabulary to fit into the context of the new language they are attempting to learn.

Frequent Trouble Spots in Training Classes

Many times the frustration starts right within the classroom itself, both in its setting and its atmosphere.

Often times I am asked what is the most difficult part of my job. Believe it or not I'd have to say it's getting familiar with all the possible terminals and keyboard configurations that there are. It seems that no two terminals and keyboards are exactly alike.

Chances are if you as the instructor are fumbling as you are helping a new user get around the system, imagine what they’re going through. Make sure as the instructor you are familiar with all the various keyboard configurations in the training room and know their caveats. Make sure adequate information is available regarding the keyboard. While a 30 minute lecture is not necessarily required, it may be helpful depending on how new the users are to the system. A few minutes up front will make them feel more comfortable and self-sufficient. Remember, chances are what they learn regarding the keyboard/terminal may change once they go back to their own terminal. Another option perhaps, is to provide written documentation next to each terminal to use as a reference should they have any questions or problems. Some sites even provide a short-course as a pre-requisite to mainframe or PC training classes.

New users to the system are going to require more of the instructor’s time during exercises for feedback and assistance. Programmers on the other hand prefer to get a problem and work through it with little or no help. As a result, a class full of new users will clamor for your time and attention and it’s your job to get to them all and not spend too much time with anyone person. This can be easily solved by discussing the exercises before the hands-on session begins so the students are clear as to what the objectives of the exercises are. As well, make sure the course materials contain copies of what the resulting output should look like to guide them in their programming process.

Maximizing the Training Experience

The instructor’s attitude alone can have major implications upon the learning experience. New users, particularly those with little or no programming background usually are hesitant about exploring and going beyond the requirements, usually out of fear of “breaking” or “ruining” something.

It is important that the instructor establishes very early a comfortable feeling within the classroom and instills in the learners a desire to “play” and explore. This can be accomplished by developing a positive psychological and physiological learning environment. Individuals are more open to learning when they feel respected. Respect their knowledge and experiences and acknowledge their importance in the learning process.

Establish a feeling of trust. People learn better from people they trust and who support them than from those people who they feel judge or threaten them. The instructor must balance criticism with praise, sympathize with their worries and concerns and be patient with them, learning a new computer language is tough stuff!

The learner needs to be and feel responsible for what they are learning. This can be easily accomplished through the course development. As mentioned previously, modularized training allows the learner to chose what they want to learn and allows for them to decide when they want to learn it. Additionally, material which provides a variety of different learning opportunities will motivate the individuals to learn at a level and a pace in which they feel most
comfortable with. Often times I like to provide a variety of hands-on exercises and I tell the class to work on the problems they find most applicable to themselves. Here again, when the responsibility is placed on the student to decide, they feel more in control of the learning process and often times will learn more as a result.

While most individuals groan when they step into a classroom where terminals have to be shared, I find in the end they don’t regret it as much as they initially thought they would. I have found for many types of adult learning, the adults provide each other with a valuable resource: support. They benefit from both teaching each other as well as learning from each other.

Four R's of Learning Programming Languages

Relevance, repetition, responsibility and retention are the four R's of training nonprogrammers. The information has to be relevant for the individual to want to learn it. Since most adults are task and problem oriented, the material will seem relevant if it will solve their problem. Modularization allows the student to decide whether or not the information is relevant and if it will assist in solving their problem. Repetition is necessary if the information is to be retained. Repetition in the form of "weaving" whereby topics, ideas and concepts are constantly referred back to and built upon generally works best in adult learning. Repetition also comes in the form of reinforcing the information through various levels and types of learning experiences. Responsibility is a major factor in determining if an adult will learn and retain any given subject. The more an adult is dictated to master something, the less likely they are to be able to retain and apply the newly learned concepts. Adults prefer to be self-directing and in control of their learning process. Again, modular training allows for them to tailor their learning process. The diversity offered in the various types and levels of learning exercises allows them to self-direct.

Ongoing Support that Won’t Tie Up Your Help Desk

Very easily new users, particularly nonprogrammers, can tie up a help desk since they lack the technical know-how, jargon and knowledge to communicate what their problems are. How do provide them with quality service without sacrificing services to the rest of the user community?

The most important part is to staff the help desk with someone who know the product and who knows it well. Most help desks have a multi-tier desk. That is one person will handle the initial call and pass it off to the appropriate person. It's important that the front-line help desk knows when to pass it off and know when they themselves can fix it. This is the part which becomes tricky with new users as when is it a JCL problem, when is it a SAS problem, or is it really a problem when the two interact? Then whose problem is it? The JCL person's or the SAS person's? This of course takes experience and a whole lot of patience on behalf of the front-line person to know what questions to ask and how to ask them.

A lot of time and headaches can be spared for the nontechnical people as well as the help desk personnel if some time is spent in putting together in-house information pertaining to the system, and which contains information and solutions to common errors encountered by new users. Providing them with copy libraries which contain skeletons or shells to build upon, particularly for JCL can really be helpful. And of course, they have to be trained or informed as to how to utilize the information and resources.

It's important particularly to supporting the nontechnical users that surveys are conducted frequently to find out where their problems exist. This can provide insight as to what additional information may be useful to them to allow them to be more self-sufficient.

Trademark Notice
SAS is a registered trademark of the SAS Institute Inc., Cary, NC, USA.

The author will be glad to answer questions and accept suggestions at the following address:

Kim L. Kolbe  
Systems Seminar Consultants  
6014 Gateway Green  
Madison, WI 53716  
Work: (608) 222-7081  
Fax: (608) 222-7310